

10/525033

PCT/JP03/10499

10 Rec'd PCT/JP 17 FEB 2005

日 本 国 特 許 庁
JAPAN PATENT OFFICE

20.08.03

REC'D 10 OCT 2003

WIPQ PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2002年 8月20日

出 願 番 号
Application Number: 特願2002-238709
[ST. 10/C]: [JP2002-238709]

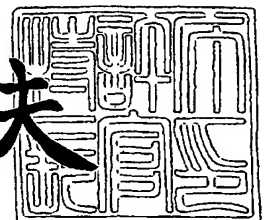
出 願 人
Applicant(s): 江良 一成

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2003年 9月25日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



Best Available Copy

【書類名】 特許願

【整理番号】 MQ02001

【提出日】 平成14年 8月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/62

【発明の名称】 立体視画像データ作成方法およびその装置

【請求項の数】 16

【発明者】

 【住所又は居所】 千葉県柏市松葉町 4-1-11-201

 【氏名】 江良 一成

【特許出願人】

 【識別番号】 596030737

 【住所又は居所】 千葉県柏市松葉町 4-1-11-201

 【氏名又は名称】 江良 一成

【代理人】

 【識別番号】 100083231

 【住所又は居所】 東京都港区新橋 2丁目 10番 5号 末吉ビル 5階 ミネ
ルバ国際特許事務所

 【弁理士】

 【氏名又は名称】 紋田 誠

【手数料の表示】

 【予納台帳番号】 016241

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9810475

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 立体視画像データ作成方法およびその装置

【特許請求の範囲】

【請求項 1】 各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、

画像データの RGB 各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出することを特徴とする立体視画像データ作成方法。

【請求項 2】 各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、

画像データの RGB 各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出する奥行き値算出ステップと、

上記二次元画像データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、

処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出され

た上記奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップを有することを特徴とする立体視画像データ作成方法。

【請求項 3】 各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、

画像データの RGB 各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出する奥行き値算出ステップと、

上記二次元画像データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、

上記二次元カラー画像データのそれぞれの画素について、1 画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する奥行き値修正ステップと、

処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記修正奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記修正奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップを有することを特徴とする立体視画像データ作成方法。

【請求項 4】 各画素の画像データが所定ビット深さを持つ二次元画像デー

タのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、

画像データのRGB各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出する奥行き値算出ステップと、

上記二次元画像データの1画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、

上記二次元カラー画像データのそれぞれの画素について、1画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第1の奥行き値修正ステップと、

上記二次元画像データの1画面に含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのおの上記奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出ステップと、

おのおのの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正ステップと、

処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の修正奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素につい

て算出された上記第2の修正奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップを有することを特徴とする立体視画像データ作成方法。

【請求項5】 各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、

上記二次元画像データの1画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、

上記二次元カラー画像データのそれぞれの画素について、1画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第1の奥行き値修正ステップと、

上記二次元画像データの1画面に含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのおのの奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出ステップと、

おのおのの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正ステップと、

おのおのの画素について、上記第2の修正奥行き値に基づき所定の丸め係数を用いて丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出ステップと、

処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値

の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行深度を加えた値を、当該画素の視差値として算出する視差値算出ステップと、

処理対象となる画素について、上記視差値算出ステップで算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成ステップよりなることを特徴とする立体視画像データ作成方法。

【請求項6】 各画素の画像データが所定ビット深さを持つ二次元画像データを入力するデータ入力ステップと、

上記二次元画像データのそれぞれの画素について、画像データ値に基づいてグレースケール値を算出するグレースケール値算出ステップと、

上記二次元画像データの1画面分のグレースケール値の平均値と、上記二次元画像データの複数サンプリング領域のグレースケール値との関係に基づいて、上記二次元画像データであらわされた1画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、上記グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断する一方、後者の場合には、上記グレースケール値の変化傾向に対して奥行き値の変化傾向が反転していると判断する奥行き値反転判断ステップと、

上記二次元画像データのそれぞれの画素について、当該画素を含む所定サイズの二次元マトリクスに含まれる上記グレースケール値の平均値を、当該画素の奥行き値として割り当てる奥行き値算出ステップと、

上記二次元カラー画像データのそれぞれの画素について上記奥行き値算出ステ

ップで算出した奥行き値の平均値と、上記奥行き値の最大値と、上記奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正して、第1の修正奥行き値を算出する第1の奥行き値修正ステップと、

上記二次元カラー画像データのそれぞれの画素について所定の丸め係数で除算して第1の丸め奥行き値を算出し、その第1の丸め奥行き値が同一値となる画素を結ぶ等高線を1つ以上作成し、その作成した等高線に基づいて、上記二次元画像データに含まれる1つ以上の独立物体を抽出し、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出し、おのこの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正ステップと、

おのこの画素について、上記第2の修正奥行き値を、所定の所定の丸め係数で除算して丸め修正奥行き値を算出し、その丸め修正奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出ステップと、

処理対象となる画素について、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を上記グレースケール値の最大値より減じた値に上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップと、

処理対象となる画素について、上記視差値算出ステップで算出された視差値、および、上記二次元カラー画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成ステップよりなることを特徴とする

立体視画像データ作成方法。

【請求項 7】 前記所定の丸め係数は、外部より入力されるオブジェクト深度で、前記グレースケール値の取りうる最大値を除算して算出されたものであることを特徴とする請求項 6 記載の立体視画像データ作成方法。

【請求項 8】 前記適用奥行き値算出ステップの前記あらかじめ設定された上限値と下限値は、それぞれ外部より入力された値であることを特徴とする請求項 6 記載の立体視画像データ作成方法。

【請求項 9】 外部より反転指示が入力されている場合には、前記視差値算出ステップでは、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第 2 の丸め奥行き値に前記奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第 2 の丸め奥行き値を上記グレースケール値の最大値より減じた値に上記奥行き深度を加えた値を、当該画素の視差値として算出することを特徴とする請求項 6 記載の立体視画像データ作成方法。

【請求項 10】 前記参照奥行き値算出ステップは、

前記二次元カラー画像データのそれぞれの画素について前記所定の丸め係数で除算して前記第 1 の丸め奥行き値を算出し、その第 1 の丸め奥行き値が同一値となる画素を結ぶ前記等高線を 1 つ以上作成する等高線数値化処理ステップと、

その作成した等高線に基づいて、上記二次元カラー画像データに含まれる 1 つ以上の前記独立物体を抽出する独立物体特定処理ステップと、

その抽出した独立物体の重心を算出し、その重心から上記二次元カラー画像データの画面の下辺に垂直に下ろした垂線と、当該独立物体に含まれる 1 つ以上の等高線の交点を算出する交点算出処理ステップと、

上記算出したおのおのの交点について、上記重心の y 座標、当該交点の y 座標、および、あらかじめ設定されている距離情報に基づいて、前記第 2 の奥行き値を算出する新奥行き値算出処理ステップと、

当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行

き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、前記第2の修正奥行き値を算出する修正奥行き値算出処理ステップからなることを特徴とする請求項6記載の立体視画像データ作成方法。

【請求項11】 前記二次元画像データは、静止画データであることを特徴とする請求項1または請求項2または請求項3または請求項4または請求項5または請求項6または請求項7または請求項8または請求項9または請求項10記載の立体視画像データ作成方法。

【請求項12】 前記二次元画像データは、動画の1フレーム分のデータであることを特徴とする請求項1または請求項2または請求項3または請求項4または請求項5または請求項6または請求項7または請求項8または請求項9または請求項10記載の立体視画像データ作成方法。

【請求項13】 前記二次元画像データは、カラー画像データであることを特徴とする請求項1または請求項2または請求項3または請求項4または請求項5または請求項6または請求項7または請求項8または請求項9または請求項10または請求項11または請求項12記載の立体視画像データ作成方法。

【請求項14】 前記二次元画像データは、白黒多階調画像データであることを特徴とする請求項1または請求項2または請求項3または請求項4または請求項5または請求項6記載または請求項7または請求項8または請求項9または請求項10または請求項11または請求項12の立体視画像データ作成方法。

【請求項15】 各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出装置において、

上記二次元画像データの1画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記二次元画像データであらわされた1画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少し

ているかを判断し、前者の場合には、上記奥行き値の変化傾向と距離の変化傾向が同一であると判断する一方、後者の場合には、上記奥行き値の変化傾向に対して距離の変化傾向が反転していると判断する奥行き値反転判断手段と、

上記二次元画像データのそれぞれの画素について、1画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第1の奥行き値修正手段と、

上記二次元画像データの1画面に含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのおのの奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出手段と、

おのおのの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正手段と、

おのおのの画素について、上記第2の修正奥行き値に基づき所定の丸め係数を用いて丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出手段と、

処理対象となる画素について、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出手段と、

処理対象となる画素について、上記視差値算出手段で算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座

標領域に保存する画像データ作成手段を備えたことを特徴とする立体視画像データ作成装置。

【請求項 16】 所定ビット深さを持つ二次元画像データを入力するデータ入力手段と、

上記二次元画像データのそれぞれの画素について、画像データ値に基づいてグレースケール値を算出するグレースケール値算出手段と、

上記二次元画像データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記二次元画像データであらわされた 1 画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、上記奥行き値の変化傾向と距離の変化傾向が同一であると判断する一方、後者の場合には、上記奥行き値の変化傾向に対して距離の変化傾向が反転していると判断する奥行き値反転判断手段と、

上記二次元画像データのそれぞれの画素について、当該画素を含む所定サイズの二次元マトリクスに含まれる上記グレースケール値の平均値を、当該画素の奥行き値として割り当てる奥行き値算出手段と、

上記二次元画像データのそれぞれの画素について上記奥行き値算出手段で算出した奥行き値の平均値と、上記奥行き値の最大値と、上記奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を修正する第 1 の奥行き値修正手段と、

上記二次元画像データのそれぞれの画素について所定の丸め係数で除算して第 1 の丸め奥行き値を算出し、その第 1 の丸め奥行き値が同一値となる画素を結ぶ等高線を 1 つ以上作成し、その作成した等高線に基づいて、上記二次元画像データに含まれる 1 つ以上の独立物体を抽出し、その抽出した独立物体に含まれる上記等高線について、新たに第 2 の奥行き値を算出し、上記第 1 の丸め奥行き値と上記第 2 の奥行き値の差を奥行き値修正値として算出する参照奥行き値算出手段と、

おのおのの画素について、上記第 1 の奥行き値修正手段で算出した奥行き値に

、上記参照奥行き値算出手段で算出した奥行き値修正値を加算して、第2の奥行き値を算出する第2の奥行き値修正手段と、

おのおのの画素について、上記第2の奥行き値修正手段で算出した第2の奥行き値を、所定の所定の丸め係数で除算して丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された範囲内の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出手段と、

処理対象となる画素について、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出手段と、

処理対象となる画素について、上記視差値算出手段で算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成手段を備えたことを特徴とする立体視画像データ作成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、写真画像や動画等、各画素の画像データが所定ビット深さを持つ二次元画像データ（白黒多階調画像データを含む）に基づいて、右目用画像データおよび左目用画像データを作成する立体視画像データ作成方法および立体視画像データ作成装置に関する。

【0002】

【従来の技術】

従来より、写真または映画を立体的な画像または映像として認識させる立体視を利用して、写真または映画を鑑賞する鑑賞方法が実用されている。

【0003】

ここで、立体視とは、『視覚、特に両眼を用いた形態視において、網膜に投影された平面像に基づく感覚情報が、大脳皮質の視覚野における統合作用を受け、三次元空間内の立体として認知されることをいう。実体視ともいわれる。立体視は、実際の立体を見る場合だけでなく、左右の目にそれぞれ一定の規則に従って異なる平面図形を見せた場合にも生ずる。立体写真や、立体図形を実体鏡で見た場合がこれにあたる。このことから、立体視には両眼の間の視差が重要であることがわかる。しかし単眼視でも、水晶体の曲率を調節している毛様体筋の収縮の程度や、目の位置を変えたときの視差などにより、ある程度の遠近や奥行き判断が可能である。霊長類では、視神経の半分が交差し、左右両眼の右半分の網膜からの神経繊維は右側皮質後頭葉の視覚野に達する。同様に、網膜の左半分が左側皮質後頭葉に連絡する。視覚野のニューロンは網膜の一定の場所に対応した入力を受けている。しかも視覚野のニューロンの大多数には、いろいろな程度に両眼からの入力があり、これが立体視に関係するといわれている。両眼視では、左右の目で見ている1つの物体は、1つに見える。眼球は、左右の網膜の像がそれぞれ対応する点にうまく合致して1つの像として見えるように反射的に運動する。これが遠近感を作る1つの原因となっている。空間知覚は、右脳に発達するので、視野の左半分の方が物体の空間的位置関係がよく認知されると言われている。』（「日本大百科全書(小学館)」より抄出）

【0004】

したがって、立体視用の2枚1組の写真を撮影するためには、同等な2つの撮影用レンズを両眼の間隔（例えば、6.5mm程度）離し、それぞれの光軸が平行になるように設置し、それぞれの撮影用レンズのシャッターを同時に作用して、2コマのフィルムに同時に映像を撮影する。そして、その2つのフィルムを現像し、印画紙に映像を焼き付けて、立体視用の写真画像を形成する。

【0005】

この2枚の立体視用写真画像を鑑賞するには、例えば、専用のビューワーを用いて2枚の立体視用写真画像を左右の目でそれぞれ鑑賞する。それにより、撮影対象が立体的に認識される。

【0006】

また、立体視用の映画を撮影するときには、所定間隔を離して設置した2台のカメラを用いて映像を撮影し、2本1組の映画フィルムを得る。そして、それぞれの映画フィルムを、偏光方向が互いに直行する2枚の偏光フィルタを映写レンズの直後に備えた2つの映写機で、同一のスクリーンに同時に映写する。

【0007】

そして、鑑賞する人は、偏光方向が互いに直行する2枚の偏光フィルタがそれぞれ両眼に対応して配設された鑑賞用眼鏡を用いてスクリーンに映写された映像を鑑賞すると、映像を立体的に認識することができる。このとき、鑑賞用眼鏡に配設される左目用フィルタおよび右目用フィルタは、左目用映像を映写する映写機の映写レンズおよび右目用映像を映写する映写機の映写レンズに設けられたと同種の偏光フィルタが配設される。

【0008】

しかしながら、このような従来の立体視用写真または立体視用映画を撮影する際には、その撮影時に、撮影する場所において、立体視用写真または立体視用映画を撮影する必要がある、時間的場所的な制約があった。

【0009】

また、静止画像から立体視画像を作成するものとしては、例えば、特開2002-123842号「立体視画像生成装置および情報記憶媒体」が提案されているものがある。この従来技術では、静止画像データの彩度に基づいて、おのおの画素について奥行き値を算出するとともに、静止画像データの明度に基づいて静止画像に含まれる物体を輪郭線抽出する。そして、その抽出した物体に含まれる画素の奥行き値の平均値を算出して、当該物体の奥行き値とし、それぞれの物体毎に算出した奥行き値を用いて、左右画像の視差を決定するようにしている。また、人の視覚特性に応じて、左右にずらした物体の周縁をぼかすようにしている。また、このぼかし処理は、元の画像データの彩度を変更することで実現している。

【0010】

また、動画（映画）の画像データについては、画面に表示される1フレームデ

ータについて、静止画データに対して行ったと同じ処理を実行することで、立体視動画データを作成することができる。

【0011】

【発明が解決しようとする課題】

しかしながら、この従来技術では、物体の一部にハイライト部分がある場合、あるいは、物体の一部が暗く落ち込んでいるような場合、その物体の一部が局部的に飛び出して見えたり、引っ込んで見えたりする場合がある。

【0012】

また、奥行き値の算出方法として、彩度を用いており、また、その根拠として、「景色は一般に遠い方向へ行くほど淡く（彩度が低下し）、近い方ほど色が鋭く（彩度が高く）なる傾向がある。遠景であれば、空気中の塵などにより彩度が低下し、物体のエッジ（輪郭線）が細く、乃至はぼやける。一方、近景では彩度が高くなる物体の数が多くなり、物体のエッジは太く、くっきりと見える。」という「自然法則」が上げられている。

【0013】

しかし、暗い森の中から遠くの明るい空が見えるような構図では、このような「自然法則」は当てはまらず、自然な立体視を得ることができないと言う不具合を生じる。

【0014】

すなわち、従来技術では、適切な奥行き値を全画素にわたって算出することが困難であり、効果的な立体視を提供できないのである。

【0015】

さて、通常の二次元画像（写真画像等）から立体視画像を自動的に作成するには、従来、次のような種々の課題がある。

【0016】

まず、従来、種々の方法が提案されているが、いずれも立体視の基礎となる視差を数値演算により算出している。しかしながら、このような視差算出の考え方は、人間の立体視感覚の一側面のみを取り上げてその基礎としており、このような方法論に問題がある。

【0017】

すなわち、このような従来方法では、人間の持つ立体視覚に関する特質を何ら考慮していないばかりか、人間が鑑賞する画像を作成するという本来の目的から逸脱しており、結果的に、自然な立体視画像を提供することができない。

【0018】

また、左右画像を認識させるための視差の算出方法は、従来では、彩度、明度あるいは色彩のいずれかに基づいて奥行き値（Z 値）を算出し、その算出した Z 値を視差と設定していた。

【0019】

例えば、上述した従来技術では、彩度を用いている。これは、上述した従来技術の場合には、上述した「自然法則」の考え方に基づくものである。しかしながら、この従来方法は、換言すれば、水墨画的遠近法に基づいて Z 値を算出しているのであって、その結果、近景の映像では Z 値を正しく算出することが困難となっている。

【0020】

さらに、このような不具合があるにもかかわらず、算出する視差を修正する等、より効果的な立体視画像を作成するための努力を怠っている。

【0021】

例えば、上述した従来技術では、エッジ算出演算により物体の輪郭を抽出した後、その抽出した輪郭を故意にボケさせ、それにより、算出した視差の不自然さが目立たないようにしている。

【0022】

これは、単に、算出した視差をごまかしているにすぎず（明細書中に記載あり）、正確な視差を算出することとは全く異なっており、自然な立体視画像を提供することはできない。

【0023】

このような不具合を解消するためには、二次元画像に含まれる物体を抽出し、その物体毎に視差を与えることにより、物体単位の立体視が可能になるとも考えられるが、上述した従来技術のように、単純なエッジ検出演算を用いて物体を抽

出すると、正確な物体の認識に失敗する。

【0024】

すなわち、二次元画像中において強い光源からの光が当たる部分では、非常に明るい部分のハイライトとなり、このハイライト部分でエッジ検出が適切に行えないと、適切な物体抽出を行うことができない。

【0025】

また、このようなハイライト部分では、Z値が大きい値として算出されるため、その部分の視差が大きくなり、いわゆる、画素クロス（隣り合う画素が異なる方向に移動すること）が発生し、画像がゆがみ、その結果、作成される立体画像は鑑賞に堪えられないほど劣化する。

【0026】

一方、単に数値的な視差を機械的に求めようとする方法では、深い立体感を得るためには、強い視差が必須となる。しかしながら、このような強い視差は、上述した画素クロスを引き起こし、画像を歪ませる原因となる。そのために、従来、弱い視差を与えて、画像のゆがみを極力小さくするようにしていた。

【0027】

そのため、従来作成された立体視画像は、例えば、表示画面から画像全体が浮いて見えるか、あるいは、沈んで見えるかのいずれかであり、画像に含まれる物体を明確に区別できるような立体視効果を得ることができなかった。

【0028】

本発明は、かかる実情に鑑みてなされたものであり、二次元画像データに基づいて、自然な立体視を得ることができるようにした立体視画像データ作成方法および立体視画像データ作成装置を提供することを目的としている。

【0029】

【課題を解決するための手段】

本発明は、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、画像データのRGB各色成分の値に、人間の網膜における赤錐体、緑錐

体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出するようにしたものである。

【0030】

また、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、画像データのRGB各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出する奥行き値算出ステップと、上記二次元画像データの1画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップを有するものである。

【0031】

また、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、画像データのRGB各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出する奥行き

値算出ステップと、上記二次元画像データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、上記二次元カラー画像データのそれぞれの画素について、1 画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する奥行き値修正ステップと、処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記修正奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記修正奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップを有するものである。

【0032】

また、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、画像データの RGB 各色成分の値に、人間の網膜における赤錐体、緑錐体および青錐体の分布に基づいて設定した重み係数を乗じ、それらの総和に基づいて算出したグレースケール値を、それぞれの画素の奥行き値として算出する奥行き値算出ステップと、上記二次元画像データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、上記二次元カラー画像データのそれぞれの画素について、1 画面分の奥行き値の平均値と、奥行き値の最

大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第1の奥行き値修正ステップと、上記二次元画像データの1画面に含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのおの上記奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出ステップと、おのおの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正ステップと、処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の修正奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の修正奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップを有するものである。

【0033】

また、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、上記二次元画像データの1画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記二次元画像データであらわされた1画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、上記奥行き値の変化傾向と距離の変化傾向が同一であると判断する一方、後者の場合には、上記奥行き値の変化傾向に対して距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、上

記二次元カラー画像データのそれぞれの画素について、1画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第1の奥行き値修正ステップと、上記二次元画像データの1画面に含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのおのの奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出ステップと、おのおのの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正ステップと、おのおのの画素について、上記第2の修正奥行き値に基づき所定の丸め係数を用いて丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出ステップと、処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップと、処理対象となる画素について、上記視差値算出ステップで算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成ステップよりなるものである。

【0034】

また、各画素の画像データが所定ビット深さを持つ二次元画像データを入力するデータ入力ステップと、上記二次元画像データのそれぞれの画素について、画像データ値に基づいてグレースケール値を算出するグレースケール値算出ステッ

ブと、上記二次元画像データの1画面分のグレースケール値の平均値と、上記二次元画像データの複数サンプリング領域のグレースケール値との関係に基づいて、上記二次元画像データであらわされた1画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、上記グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断する一方、後者の場合には、上記グレースケール値の変化傾向に対して奥行き値の変化傾向が反転していると判断する奥行き値反転判断ステップと、上記二次元画像データのそれぞれの画素について、当該画素を含む所定サイズの二次元マトリクスに含まれる上記グレースケール値の平均値を、当該画素の奥行き値として割り当てる奥行き値算出ステップと、上記二次元カラー画像データのそれぞれの画素について上記奥行き値算出ステップで算出した奥行き値の平均値と、上記奥行き値の最大値と、上記奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正して、第1の修正奥行き値を算出する第1の奥行き値修正ステップと、上記二次元カラー画像データのそれぞれの画素について所定の丸め係数で除算して第1の丸め奥行き値を算出し、その第1の丸め奥行き値が同一値となる画素を結ぶ等高線を1つ以上作成し、その作成した等高線に基づいて、上記二次元画像データに含まれる1つ以上の独立物体を抽出し、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出し、おのこの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正ステップと、おのこの画素について、上記第2の修正奥行き値を、所定の所定の丸め係数で除算して丸め修正奥行き値を算出し、その丸め修正奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出ステップと、処理対象となる画素について、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断されてい

る場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を上記グレースケール値の最大値より減じた値に上記奥行深度を加えた値を、当該画素の視差値として算出する視差値算出ステップと、処理対象となる画素について、上記視差値算出ステップで算出された視差値、および、上記二次元カラー画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成ステップよりなる立体視画像データ作成方法である。

【0035】

また、前記所定の丸め係数は、外部より入力されるオブジェクト深度で、前記グレースケール値の取りうる最大値を除算して算出されたものである。

【0036】

また、前記適用奥行き値算出ステップの前記あらかじめ設定された上限値と下限値は、それぞれ外部より入力された値である。

【0037】

また、外部より反転指示が入力されている場合には、前記視差値算出ステップでは、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に前記奥行深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記グレースケール値の変化傾向と奥行き値の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を上記グレースケール値の最大値より減じた値に上記奥行深度を加えた値を、当該画素の視差値として算出する。

【0038】

また、前記参照奥行き値算出ステップは、前記二次元カラー画像データのそれぞれの画素について前記所定の丸め係数で除算して前記第1の丸め奥行き値を算出し、その第1の丸め奥行き値が同一値となる画素を結ぶ前記等高線を1つ以上

作成する等高線数値化処理ステップと、その作成した等高線に基づいて、上記二次元カラー画像データに含まれる1つ以上の前記独立物体を抽出する独立物体特定処理ステップと、その抽出した独立物体の重心を算出し、その重心から上記二次元カラー画像データの画面の下辺に垂直に下ろした垂線と、当該独立物体に含まれる1つ以上の等高線の交点を算出する交点算出処理ステップと、上記算出したおのおのの交点について、上記重心のy座標、当該交点のy座標、および、あらかじめ設定されている距離情報に基づいて、前記第2の奥行き値を算出する新奥行き値算出処理ステップと、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、前記第2の修正奥行き値を算出する修正奥行き値算出処理ステップからなる。

【0039】

また、前記二次元画像データは、静止画データである。また、前記二次元画像データは、動画の1フレーム分のデータである。また、前記二次元画像データは、カラー画像データである。また、前記二次元画像データは、白黒多階調画像データである。

【0040】

また、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出装置において、上記二次元画像データの1画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記二次元画像データであらわされた1画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、上記奥行き値の変化傾向と距離の変化傾向が同一であると判断する一方、後者の場合には、上記奥行き値の変化傾向に対して距離の変化傾向が反転していると判断する奥行き値反転判断手段と、上記二次元画像データのそれぞれの画素について、1画面分の奥行き値の平均値と、奥

行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第1の奥行き値修正手段と、上記二次元画像データの1画面に含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのこの奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出手段と、おのこの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第2の修正奥行き値を算出する第2の奥行き値修正手段と、おのこの画素について、上記第2の修正奥行き値に基づき所定の丸め係数を用いて丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第2の丸め奥行き値を算出する適用奥行き値算出手段と、処理対象となる画素について、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出手段と、処理対象となる画素について、上記視差値算出手段で算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成手段を備えたものである。

【0041】

また、所定ビット深さを持つ二次元画像データを入力するデータ入力手段と、上記二次元画像データのそれぞれの画素について、画像データ値に基づいてグレースケール値を算出するグレースケール値算出手段と、上記二次元画像データのそれぞれの画素について算出したグレースケール値の平均値と、上記二次元画像

データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記二次元画像データであらわされた 1 画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、上記奥行き値の変化傾向と距離の変化傾向が同一であると判断する一方、後者の場合には、上記奥行き値の変化傾向に対して距離の変化傾向が反転していると判断する奥行き値反転判断手段と、上記二次元画像データのそれぞれの画素について、当該画素を含む所定サイズの二次元マトリクスに含まれる上記グレースケール値の平均値を、当該画素の奥行き値として割り当てる奥行き値算出手段と、上記二次元画像データのそれぞれの画素について上記奥行き値算出手段で算出した奥行き値の平均値と、上記奥行き値の最大値と、上記奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を修正する第 1 の奥行き値修正手段と、上記二次元画像データのそれぞれの画素について所定の丸め係数で除算して第 1 の丸め奥行き値を算出し、その第 1 の丸め奥行き値が同一値となる画素を結ぶ等高線を 1 つ以上作成し、その作成した等高線に基づいて、上記二次元画像データに含まれる 1 つ以上の独立物体を抽出し、その抽出した独立物体に含まれる上記等高線について、新たに第 2 の奥行き値を算出し、上記第 1 の丸め奥行き値と上記第 2 の奥行き値の差を奥行き値修正値として算出する参照奥行き値算出手段と、おのこの画素について、上記第 1 の奥行き値修正手段で算出した奥行き値に、上記参照奥行き値算出手段で算出した奥行き値修正値を加算して、第 2 の奥行き値を算出する第 2 の奥行き値修正手段と、おのこの画素について、上記第 2 の奥行き値修正手段で算出した第 2 の奥行き値を、所定の所定の丸め係数で除算して丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された範囲内の値に制限して第 2 の丸め奥行き値を算出する適用奥行き値算出手段と、処理対象となる画素について、上記奥行き値反転判断手段で上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第 2 の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断手段で上記奥行き値の変化傾

向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第2の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出手段と、処理対象となる画素について、上記視差値算出手段で算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成手段を備えたものである。

【0042】

【発明の実施の形態】

以下、添付図面を参照しながら、本発明の実施の形態を詳細に説明する。

【0043】

まず、立体視を効果的に実現できるための工夫の背景について説明する。

【0044】

立体視は、両眼を備える人間が、左右それぞれの目で見える映像が別々に脳に到達し、処理されることで認識される感覚の1つである。

【0045】

しかしながら、我々が普通の映像（通常の二次元映像等の非立体視映像）を見ても遠近感や立体感を感知できており、映像を楽しむことができる。

【0046】

これらのことから、立体視には、視覚からの映像の処理だけでなく、映像を認識する際の学習効果からも大きい影響があるものと考えられる。例えば、だまし絵の多くは、立体視に関する学習効果を逆手に取り、見ている人が立体的な効果を体得できるような表現をしたものである。

【0047】

本発明では、単に視覚から受ける視差のみで立体感を構成するのではなく、1物体上のZ値（奥行き値）の変化を滑らかにし、物体と背景との境界の輪郭部のZ値の変化にメリハリを付けることで、より自然な立体感を感じるように工夫している。

【0048】

一般に、普通の視力をもつ人間が、視覚により立体感を識別できるのは、約 140 (メートル) 程度である。我々は、風景を見ると、立体視できる映像 (近景) と、立体視できない映像 (遠景) の混在した映像で、遠近感を感知していることとなる。

【0049】

ところが、かかる事情があっても、通常の状態では、映像に違和感がなく、風景を認識している。

【0050】

すなわち、140 (メートル) を超えた遠景の部分については、われわれは、立体に対する学習効果に基づいて、立体を認識していると考えられる。

【0051】

このようなことを背景に、本発明は、人間が効果的に立体視を感知できるような立体視画像データを作成する方法を工夫している。

【0052】

すなわち、換言すれば、本発明は、単なる数値演算結果により視差を求めるだけではなく、人間が強く立体感を感じる場合にはどのような条件が必要であるかの研究に基づいてなされたものである。

【0053】

かかる研究によれば、物体と物体とが重なり合う部分に視差の違いが顕著に存する場合に、人間はより強く立体感を知覚できるとの結論に達した。

【0054】

しかしながら、物体と物体が重なり合う部分の多くには、強い光源からの光が当たり、ハイライト部分が形成されて、強い視差が形成され、その結果、物体と物体の重なり合う部分に画像の乱れが生じやすく、深い視差を与えるには限界があった。

【0055】

このような限界を乗り越え、より深い視差を求めるためには、画像の乱れの原因となる局所的な強い Z 値や極端に低い Z 値を、違和感や立体感の低下が生じないように補正できれば良いこととなる。

【0056】

そのため、本発明は、次のような種々の補正手段を創出した。

【0057】

①暗部におけるZ値の拡張、②輪郭抽出、③丸め係数、④画像の特質の判定、⑤上限基準・下限基準による高Z値・低Z値の補正、⑥メッシュZ値によるZ値順序の補正。

【0058】

図1は、本発明の一実施例にかかる立体視用画像データ作成装置の一例を示している。

【0059】

同図において、CPU（中央処理装置）1は、この立体視用画像データ作成装置の動作制御、立体視用画像データ作成処理などの各種処理を実行するためのものであり、ROM（リード・オンリ・メモリ）2は、起動時などに実行されるプログラムや参照データなどが記憶されるものであり、RAM（ランダム・アクセス・メモリ）3は、CPU1のワークエリアなどを実現するためのものであり、キャラクタジェネレータ4は、図形文字の表示データ（フォントデータ）を発生するものであり、時計回路5は、現在時刻情報を出力するためのものである。

【0060】

インターネットインターフェース回路6は、この立体視用画像データ作成装置をインターネット（図示略）へ接続するためのものであり、インターネット伝送制御部7は、インターネットを介して、他の端末装置との間で種々のデータをやりとりするための各種所定のプロトコルスイートの通信制御処理を実行するためのものである。

【0061】

磁気ディスク装置8は、この立体視用画像データ作成装置で実行される各種処理やオペレーティングシステムなどのプログラムデータを記憶するとともに、各種処理により作成される各種データなどを記憶する大容量の外部記憶装置であり、CD-ROM装置9は、CD-ROM10に記憶されているデータを再生するためのものである。

【0062】

CRT画面表示装置11は、カラー画像データを表示するためのものであり、表示制御部12は、CRT画面表示装置11の表示内容を制御するためのものである。

【0063】

キーボード装置13は、フルキー配列を備えて、ユーザがキー操作を行うためのものであり、画面指示装置14は、ユーザが表示画面をクリック操作するなどの種々の操作を行うためのものであり、入力制御部15は、キーボード装置13および画面指示装置14の入力データを取り込むためのものである。

【0064】

これらのCPU1、ROM2、RAM3、キャラクタジェネレータ4、時計回路5、インターネット伝送制御部7、磁気ディスク装置8、CD-ROM装置9、表示制御部12、および、入力制御部15は、内部バス16に接続されており、これらの各種要素間のデータのやりとりは、主としてこの内部バス16を介して行われている。

【0065】

さて、本実施例では、立体視画像表示用のアプリケーションにより、立体視用画像データが作成される。その立体視画像表示アプリケーションの表示画面の一例を図2(a)、(b)に示す。

【0066】

この表示画面は、メイン画面MDとメニュー項目が配列されるメニューバーMBからなり、同図(a)の表示画面は、ユーザが適宜に表示選択した動画データがメイン画面MDに表示されている。

【0067】

また、メニューバーMBに表示されている項目MI1は、ユーザが処理対象となる動画データの選択等を行うための「ファイルメニュー」を選択するためのものであり、項目MI2は、ユーザが立体視画像を表示する際の設定操作を行うためのものであり、項目MI3は、立体視画像データを記録する際にユーザが選択操作するためのものであり、項目MI4は、この立体視画像表示アプリケーション

ンを終了させる操作を行うためのものであり、項目MI 5は、メイン画面MDに表示している動画の再生を行う旨を操作するためのものであり、項目MI 6は、メイン画面MDに表示している動画の停止を行う旨を操作するためのものであり、項目MI 7は、メイン画面MDに表示している動画の一時停止を行う旨を操作するためのものであり、項目MI 8は、メイン画面MDに表示している動画の表示モードが通常モード（2D）であるかあるいは立体視モード（3D）であるかの表示とその表示モードの切換を指定するための操作を行うためのものであり、項目MI 9は、立体視表示モードの種類（Odd-evenモード／余色モード）の選択操作を行うためのものであり、項目MI 10は、立体視において飛び出して見える方向の切換操作を行うためのものである。

【0068】

また、同図（b）の表示画面は、立体視モードの場合のものであり、この場合、メイン画面MDには、立体視画像データが表示される。

【0069】

また、本実施例では、項目MI 8をユーザが操作するたびに、通常モードと立体視モードが切換えられるようになっている（トグル動作）。

【0070】

図3は、項目MI 2を選択操作した場合に表示される設定ダイアログパネルの一例を示している。

【0071】

この設定ダイアログパネルは、上部にタイトルバーBRが表示され、そのタイトルバーには、この設定ダイアログパネルを閉じるためのクローズボックスBCが配置されている。

【0072】

また、設定ダイアログパネルには、奥行深度（後述）を入力するためのスライダバーIS 1、スライダバーIS 1により入力された奥行深度を表示するとともに適宜に値を入力するための入力窓IB 1、オブジェクト深度（後述）を入力するためのスライダバーIS 2、スライダバーIS 2により入力されたオブジェクト深度を表示するとともに適宜に値を入力するための入力窓IB 2、奥行き値（

後述) の上限値を入力するための入力窓 I B 3、奥行き値の下限值を入力するための入力窓 I B 4、奥行き値の上限値および下限値をデフォルト値に設定するための操作ボタン B M 1、開始ラインを奇数ラインとするか偶数ラインとするかを設定するための入力ボックス I B 5、I B 6、この設定ダイアログパネルの設定内容を処理へ反映することを指定操作するための O K ボタン B M 2、および、この設定ダイアログパネルの設定内容を処理へ反映しないことを指定操作するためのキャンセルボタン B M 3 が、それぞれ適宜に配置されている。

【0073】

ここで、「奥行深度」とは、画像表示のインタフェース(画面)から立体世界が、どの位置に表示されるかの深さを示す。例えば、左目画像について視差分を左に移動し、右目画像について視差分を右に移動して表示された立体視画像が、表示画面から内部に引いて見えている状態であるとすれば、視差分の移動方向を逆転することで、立体視画像は、表示画面から飛び出して見える。

【0074】

また、「オブジェクト深度」とは、立体視世界の最低奥行から最高奥行までの深さを示す。このパラメータの値を大きくすると、立体感が増加することとなる。

【0075】

また、「奥行き値」とは、画像に含まれる物体に割り当てる(画面に対する)奥行方向の距離をあらわしている。

【0076】

また、この設定ダイアログパネルにおける各種入力値や、設定値は、後述する処理において参照される。

【0077】

また、本実施例では、C R T 表示装置 1 1 の表示画面に、偏光ストライプフィルタを貼り付け、ユーザは、偏光メガネを装着して表示画面を観察することで、立体視画像を鑑賞できるようにしている。このような立体視表示方式は、O d d - E v e n 方式という。

【0078】

すなわち、まず、図 4 に示すように、C R T 表示装置 1 1 の表示画面の偶数ライン S L 0, S L 2, . . . には、左目用の偏光ストライプフィルタ F L 0, F L 2, . . . が、それぞれに対応した幅と間隔で貼り付けられるとともに、C R T 表示装置 1 1 の表示画面の奇数ライン S L 1, S L 3, . . . には、右目用の偏光ストライプフィルタ F L 1, F L 3, . . . が、それぞれに対応した幅と間隔で貼り付けられる。

【0079】

そして、図 5 に示すように、立体視用画像データのうちの左目用画像データを C R T 表示装置 1 1 の偶数ライン S L 0, S L 2, . . . に表示するとともに、右目用画像データを C R T 表示装置 1 1 の奇数ライン S L 1, S L 3, . . . に表示する。

【0080】

ユーザは、偏光メガネ G S を装着してこの C R T 表示装置 1 1 の表示画面を見ると、C R T 表示装置 1 1 の偶数ライン S L 0, S L 2, . . . の表示内容は偏光ストライプフィルタ F L 0, F L 2, . . . を通過するので、左目に対応して配設されたフィルタ F L f を透過して左目に到達するとともに、右目に対応して配設されたフィルタ F L r には遮断される。

【0081】

一方、C R T 表示装置 1 1 の奇数ライン S L 1, S L 3, . . . の表示内容は偏光ストライプフィルタ F L 1, F L 3, . . . を通過するので、フィルタ F L r を透過して右目に到達するとともに、フィルタ F L f には遮断される。

【0082】

それにより、ユーザの左目では左目用画像が、また、右目では右目用画像がそれぞれ感知され、その結果、ユーザは、表示画像を立体視する。

【0083】

図 6 は、立体視画像表示アプリケーションにおいて、立体視表示モードが選択された場合に行われる処理の一例を示している。

【0084】

まず、カラー動画データの 1 フレーム分のデータ（カラー画像データ（＝二次

元カラー画像データ)) を読み込み (処理 101)、そのカラー画像データが RGB フォーマットのものであるかどうかを調べる (判断 102)。

【0085】

判断 102 の結果が NO になるとときには、処理 101 で読み込んだ 1 フレーム分のカラー画像データを、RGB フォーマットのカラー画像データへ変換する (処理 103)。

【0086】

ここで、カラー動画データとしては、任意の圧縮フォーマットのものを採用することができる。その場合、処理 101 で読み込む 1 フレーム分のカラー画像データは、カラー動画データを復号化して得られた 1 フレーム分の表示画像データである。すなわち、本実施例における処理対象は、CRT 表示装置 11 に表示させる 1 フレーム分のカラー画像データである。また、カラー動画データの 1 画素あたりの色彩深度としては、16 ビット、24 ビット、または、32 ビットのいずれかのものを適用することができる。

【0087】

また、カラー動画データは、磁気ディスク装置 8 や CD-ROM 10 に保存されているものであってもよいし、また、インターネットより受信したものでよい。

【0088】

そして、その 1 フレーム分のカラー画像データについて、まず、それぞれの画素について、画像データ値に基づいてグレースケール値を算出し、その算出したグレースケール値を Z 値 (奥行き値) とする Z 値算出処理 (処理 104 (=グレースケール値算出ステップ)) を行う。

【0089】

次に、1 フレーム分のカラー画像データのそれぞれの画素について算出したグレースケール値の平均値と、カラー画像データの複数サンプリング領域のグレースケール値との関係に基づいて、そのフレームの画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部

から周辺部へ向かって減少しているかを判断し、前者の場合には、グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断する一方、後者の場合には、グレースケール値の変化傾向に対して奥行き値の変化傾向が反転していると判断する Z 値反転判断処理（処理 105（＝奥行き値反転判断ステップ））を行う（上述した④に関連する処理を含む）。

【0090】

次いで、カラー画像データのそれぞれの画素について、当該注目画素（処理対象となっている画素；以下同じ）を含む所定サイズの二次元マトリクスに含まれるグレースケール値の平均値を、当該画素の奥行き値として割り当てる輪郭抽出基礎データ処理（処理 106（＝奥行き値算出ステップ））を行う（上述した②に関する処理を含む）。

【0091】

次に、カラー画像データのそれぞれの画素について処理 106 で算出した奥行き値の画面全体の平均値と、最大値と、最小値に基づいて、それぞれの画素について、奥行き値を修正する Z 値拡張処理（処理 107（＝第 1 の奥行き値修正ステップ））を行う（上述した①に関する処理）。

【0092】

次に、カラー画像データに含まれる 1 つ以上の独立物体を抽出し、その抽出した独立物体に含まれる等高線（後述）について、新たに奥行き値を算出し、その新たに算出した奥行き値に基づいて、奥行き値修正値を算出する参照 Z 値算出処理（処理 108（＝参照奥行き値算出ステップ））を行う（上述した③、⑥に関する処理を含む）。

【0093】

次いで、おのおのの画素について、処理 108 で算出した奥行き値に、上記参照奥行き値算出ステップで算出した奥行き値修正値を加算して、奥行き値を修正する Z 値修正処理（処理 109（＝第 2 の奥行き値修正ステップ））を行う。

【0094】

次に、おのおのの画素について、それまでに算出して得られたデータに基づいて、左目用画像データおよび右目用画像データを作成して出力する立体変換出力

処理（処理 110（＝適用奥行き値算出ステップ、視差値算出ステップ、および、画像データ作成ステップ））を行う。

【0095】

次に、各処理について具体的に説明する。なお、以下の説明において、式の構文やデータ型等の記述方法は、C言語またはその拡張言語（C++言語等）の記述方法に準拠している。

【0096】

上述したように、本実施例では、色彩深度として16ビット、24ビット、および、32ビットのいずれかのRGBフォーマットのカラー画像データを用いることができる。

【0097】

ここで、色彩深度が16ビットのカラー画像データの1画素のデータは、図7(a)に示すように、Rデータが5ビット、Gデータが6ビット、および、Bデータが5ビットのデータ構成を持つ。

【0098】

また、色彩深度が24ビットのカラー画像データの1画素のデータは、同図(b)に示すように、Rデータが8ビット（＝1バイト）、Gデータが8ビット、および、Bデータが8ビットのデータ構成を持つ。

【0099】

また、色彩深度が32ビットのカラー画像データの1画素のデータは、同図(c)に示すように、24ビットのカラー画像データの先頭に8ビットのダミーデータを付加したデータ構成を持つ。

【0100】

図8, 9, 10は、Z値算出処理（処理104）の一例を示している。

【0101】

このZ値算出処理では、それぞれの画素について、Rデータの30%と、Gデータの59%と、Bデータの11%の総和を取った値を、8ビットのグレースケール値として算出し、そのグレースケール値を当該画素のZ値（zinfo）として算出している。なお、Rデータ、Gデータ、および、Bデータの重み付けの

値（30%、59%、11%）は、人間の網膜に存在する赤錐体、緑錐体、および、青錐体の数の分布に基づき、立体視画像を作成する際に特に適する値を設定したものである。

【0102】

ここで、色彩深度が16ビットの場合には、グレースケール値の最大値は31であり、色彩深度が24ビットおよび32ビットの場合には、グレースケール値の最大値は255である。

【0103】

ここで、*srcは元画像データが格納された領域へのポインタ、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、*zinfoは計算されたZ値を格納する領域へのポインタをあらわす。

【0104】

まず、色彩深度が16ビットであるか否かを調べる（判断201）。判断201の結果がYESになるときは、「Unsigned Short *srcp」を変数宣言し（処理202）、注目画素のデータsrcをunsigned shortへ型変換し（処理203）、変数srcpに、データsrcを代入する（処理204）。

【0105】

次いで、カウンタCCの値を0に初期設定し（処理205）、カウンタCCの値が、「WD*HT」よりも小さいかどうかを調べる（判断206）。判断206の結果がNOになるときは、1画面分の画素についての処理が終了したので、この処理を終了する。

【0106】

また、判断206の結果がYESになるときは、1画面分の画素についての処理が終了していない場合である。

【0107】

この場合には、変数Rに、ポインタ*srcpに格納されているデータ（16ビットデータ）と16進数「7c00」を論理積した結果を10ビット算術右シフトし、その結果に30を乗じた値を、intに型変換した結果の値を代入する

(処理207)。それにより、変数Rには、Rデータ(5ビット)に30を乗じた値が代入される。

【0108】

次に、変数Gに、ポインタ*srcpに格納されているデータ(16ビットデータ)と16進数「03e0」を論理積した結果を5ビット算術右シフトし、その結果に59を乗じた値を、intに型変換した結果の値を代入する(処理208)。それにより、変数Gには、Gデータ(6ビット)に59を乗じた値が代入される。

【0109】

次に、変数Bに、ポインタ*srcpに格納されているデータ(16ビットデータ)と16進数「001f」を論理積した結果に11を乗じた値を、intに型変換した結果の値を代入する(処理209)。それにより、変数Bには、Bデータ(5ビット)に11を乗じた値が代入される。

【0110】

そして、変数Rと変数Gと変数Bの値の総和を100で除算した結果を、ポインタ*zinfoであらわされる領域へ代入する(処理210)。

【0111】

これで、1画素分の処理が終了したので、変数srcp, zinfo、カウンタCCの値をそれぞれ1つ増やして(処理211)、判断206へ戻り、それ以降の処理を行う。

【0112】

また、判断201の結果がNOになるときには、色彩深度が24ビットであるか否かを調べる(判断215)。判断215の結果がYESになるときには、カウンタCCの値を0に初期設定し(処理216)、カウンタCCの値が、「WD*HT」よりも小さいかどうかを調べる(判断217)。判断217の結果がNOになるときには、1画面分の画素についての処理が終了したので、この処理を終了する。

【0113】

また、判断217の結果がYESになるときには、1画面分の画素についての

処理が終了していない場合である。

【0114】

この場合には、変数Bに、ポインタ*srcに格納されているデータに11を乗じた値を代入する(処理218)。次に、変数Gに、ポインタ*(src+1)に格納されているデータに59を乗じた値を代入する(処理219)。次いで、変数Rに、ポインタ*(src+2)に格納されているデータに30を乗じた値を代入する(処理220)。

【0115】

そして、変数Rと変数Gと変数Bの値の総和を100で除算した結果を、ポインタ*zinfoであらわされる領域へ代入する(処理221)。

【0116】

これで、1画素分の処理が終了したので、変数srcp, zinfo、カウンタCCの値をそれぞれ1つ増やして(処理222)、判断217へ戻り、それ以降の処理を行う。

【0117】

また、判断215の結果がNOになるときには、色彩深度が32ビットの場合である。この場合には、「Unsigned long *srcp」を変数宣言し(処理225)、注目画素のデータsrcをunsigned longへ型変換し(処理226)、変数srcpに、データsrcを代入する(処理227)。

【0118】

次いで、カウンタCCの値を0に初期設定し(処理228)、カウンタCCの値が、「WD*HT」よりも小さいかどうかを調べる(判断229)。判断229の結果がNOになるときには、1画面分の画素についての処理が終了したので、この処理を終了する。

【0119】

また、判断229の結果がYESになるときには、1画面分の画素についての処理が終了していない場合である。

【0120】

この場合には、変数Rに、ポインタ*srcpに格納されているデータ（32ビットデータ）と16進数「00ff0000」を論理積した結果を16ビット算術右シフトし、その結果に30を乗じた値を、intに型変換した結果の値を代入する（処理230）。それにより、変数Rには、Rデータ（8ビット）に30を乗じた値が代入される。

【0121】

次に、変数Gに、ポインタ*srcpに格納されているデータ（32ビットデータ）と16進数「0000ff00」を論理積した結果を5ビット算術右シフトし、その結果に59を乗じた値を、intに型変換した結果の値を代入する（処理231）。それにより、変数Gには、Gデータ（8ビット）に59を乗じた値が代入される。

【0122】

次に、変数Bに、ポインタ*srcpに格納されているデータ（32ビットデータ）と16進数「0000ff」を論理積した結果に11を乗じた値を、intに型変換した結果の値を代入する（処理232）。それにより、変数Bには、Bデータ（8ビット）に11を乗じた値が代入される。

【0123】

そして、変数Rと変数Gと変数Bの値の総和を100で除算した結果を、ポインタ*zinfoであらわされる領域へ代入する（処理233）。

【0124】

これで、1画素分の処理が終了したので、変数srcp, zinfo、カウンタCCの値をそれぞれ1つ増やして（処理234）、判断228へ戻り、それ以降の処理を行う。

【0125】

次に、Z値反転判定処理（処理105）について説明する。

【0126】

このZ値反転判定処理では、立体視をした際に、Z値が大きいほど、遠くにあるような立体感を得ることができるようにするため、カラー画像の傾向を調べ、Z値を反転修正するか否かを判定している。

【0127】

まず、図11に示すように、カラー画像データの1画面の左上隅、中央上端、右上隅、中央左端、中央、中央右端、左下隅、中央下端、右下隅に、それぞれ縦8ドットで横8ドットのサイズの二次元マトリクス領域を設け、それぞれセンサSS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8, SS9と名付ける。なお、これらのセンサSS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8, SS9を構成する二次元マトリクス領域の大きさは、縦8ドットで横8ドットに限らず、適宜な大きさを設定することができる。

【0128】

そして、おのおののセンサSS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8, SS9に含まれる画素のZ値の平均値を、それぞれのセンサSS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8, SS9のZ値として用いる。

【0129】

そして、それぞれのセンサSS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8, SS9には、重み値として「3」、「4」、「3」、「2」、「5」、「2」、「1」、「2」、「1」をそれぞれ割り当てる。

【0130】

この場合、各センサへの重み値の割り当て規則は、「中央が最も大きく、かつ、上から下方向に向かって小さく」というものである。

【0131】

なお、このセンサの個数は、9個に限らず、例えば、16個に設定することができる。

【0132】

図12, 13は、Z値反転判定処理の一例を示している。

【0133】

まず、カラー画像データの全画素のZ値の平均値MZを算出し（処理301）、変数S1, S2、および、カウンタCTの値を0に初期設定する（処理302）。

【0134】

次いで、カウンタCTの値が9より小さいかどうかを調べ（判断303）、判断303の結果がYESになるとときには、CT番目のセンサSS（SS（CT））のZ値が、平均値MZ以上であるかどうかを調べる（判断304）。

【0135】

判断304の結果がYESになるとときには、変数S2に、CT番目のセンサSSの重み値を積算する（処理305）。また、判断304の結果がNOになるとときには、変数S1に、CT番目のセンサSSの重み値を積算する（処理306）。

【0136】

これにより、変数S1には、画面の明るさの平均値よりも暗いセンサの重み値が加算され、変数S2には、画面の明るさの平均値よりも明るいセンサの重み値が加算される。

【0137】

次に、カウンタCTの値を1つ増やし（処理307）、判断303に戻る。

【0138】

判断303の結果がNOになるとときには、変数S1の値が、変数S2の値よりも大きいかどうかを調べる（判断308）。判断308の結果がYESになるとときには、Z値を反転する（意味については後述）と判定する（処理309）。

【0139】

また、判断308の結果がNOになるとときには、変数S2の値が、変数S1の値よりも大きいかどうかを調べる（判断310）。判断310の結果がYESになるとときには、Z値を反転しない（意味については後述）と判定する（処理311）。

【0140】

また、判断310の結果がNOになるとときには、変数S1と変数S2の値が等しい場合なので、中央部のセンサSS5のZ値が、平均値MZ以上であるかどうかを調べる（判断312）。

【0141】

判断 3 1 2 の結果が Y E S になるときには、Z 値を反転しないと判定し（処理 3 1 3）、判断 3 1 2 の結果が Y E S になるときには Z 値を反転すると判定する（処理 3 1 4）。

【0 1 4 2】

次いで、立体視画像表示アプリケーションの表示画面の項目 M I 1 0 の F 1 / F 2 モードの設定が、F 2 モードになっているかどうかを調べる（判断 3 1 5）

。

【0 1 4 3】

判断 3 1 5 の結果が Y E S になるときには、そのときの判定結果を反転する（処理 3 1 6）。すなわち、Z 値を反転すると判定している場合には、Z 値を反転しないと最終的に決定し、また、Z 値を反転しないと判定している場合には、Z 値を反転すると最終的に決定する。

【0 1 4 4】

そして、この処理を終了する。

【0 1 4 5】

次に、輪郭抽出基礎データ処理（処理 1 0 6）について説明する。

【0 1 4 6】

この輪郭抽出基礎データ処理では、図 1 4 に示すように、各画素について、当該画素位置を左上隅に配置した縦 1 2 ドット（M y）で横 1 2 ドット（M x）の二次元マトリクス領域 M M を抽出し、その二次元マトリクス領域 M M に含まれる 1 4 4 画素の Z 値の平均値を算出して、その平均値を、注目画素 P x の Z 値（o u t i n f o）として割り当てる。なお、この二次元マトリクス領域 M M の大きさは、縦 1 2 ドットで横 1 2 ドットに限ることはなく、適宜な大きさを設定することができる。

【0 1 4 7】

また、x 座標が（W D - 1 2）よりも小さい値の注目画素 P x、および、y 座標が（H T - 1 2）よりも小さい値の注目画素 P x については、二次元マトリクス領域 M M に含まれる画素のデータのみを抽出して、平均値を算出し、その算出した平均値を注目画素 P x の Z 値（o u t i n f o）として割り当てる。

【0148】

この輪郭抽出基礎データ処理では、立体視した際の画像の乱れが抑制され、また、画像の輪郭が鮮明になると言う効果が得られる。

【0149】

これらの効果について説明する。まず、立体視するためには、各画素について元の位置から視差に相当する量を左右いずれかに移動する必要がある、その移動量は、Z値から求めている。このために、Z値の変化、すなわち、視差の変化が激しい部分では、画素のクロス（隣接する画素が、それぞれ左右別方向へ移動すること）が発生する。このような画素のクロスは、物体に光が当たってハイライトになっている輪郭の部分などに多く発生し、その結果、輪郭のぼやけ、にじみ、干渉縞模様などが、立体視した画像に出現する。したがって、立体視画像を生成する際には、このような画素のクロス現象が発生しない範囲で視差を決定することが重要である。

【0150】

本実施例の輪郭抽出基礎データ処理では、二次元マトリクス領域MMの範囲で平均値を算出して、注目画素P_xのZ値として割り当てるようにしているので、極端にZ値が変化している部分のZ値の変化値がなめらかになり、その結果、画素のクロス現象を抑制することができる。

【0151】

すなわち、輪郭抽出基礎データ処理により、画像の乱れが抑制されるとともに、画像の輪郭が鮮明になる。

【0152】

また、画像に含まれる複数物体について、同一物体のZ値の変化を抑制しているので、物体が重なっている部分では、その物体の境界を構成する輪郭が際だつこととなり、結果的に、おのこの物体の独立性が保持され、その部分での物体の区別が強調される。

【0153】

図15は、輪郭抽出基礎データ処理（処理106）の一例を示している。

【0154】

ここで、`*zinfo`は計算されたZ値が格納された領域へのポインタ、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、`*outinfo`は輪郭抽出基礎データを格納する領域へのポインタをあらわす。

【0155】

まず、カウンタLNを0に初期設定し（処理401）、カウンタLNの値が縦幅画素数HTより小さいかどうかを調べる（判断402）。判断402の結果がNOになるときは、1画面分の処理が終了したので、この処理を終了する。

【0156】

また、判断402の結果がYESになるときは、 $(HT-LN)$ の値が12以上であるかどうかを調べ（判断403）、判断403の結果がYESになるときは、二次元マトリクス領域MMの縦方向のサイズを記憶するための変数MThに12を代入し（処理404）、判断403の結果がNOになるときは、変数MThに $(HT-LN)$ の値を代入する（処理405）。

【0157】

次に、カウンタCMの値を0に初期設定し（処理406）、カウンタCMの値が横幅画素数WDより小さいかどうかを調べる（判断407）。判断407の結果がYESになるときは、 $(WD-CM)$ の値が12以上であるかどうかを調べ（判断408）、判断408の結果がYESになるときは、二次元マトリクス領域MMの幅方向のサイズを記憶するための変数MTwに12を代入し（処理409）、判断408の結果がNOになるときは、変数MTwに $(WD-CM)$ の値を代入する（処理410）。

【0158】

次いで、そのときのカウンタLN、CMの値で示される注目画素について変数MTh、MTwで示される二次元マトリクス領域MMに含まれる全画素のZ値（`zinfo`）の合計値SM1を算出し（処理411）、注目画素のZ値として、`outinfo`に $SM1 / (MTh * MTw)$ の値（すなわち、二次元マトリクス領域MMのZ値の平均値）を格納する（処理412）。

【0159】

次いで、カウンタCMの値を1つ増やして（処理413）、判断407へ戻る

【0160】

また、判断407の結果がNOになるとときには、1ライン分の画素についての処理が終了したので、カウンタLNの値を1つ増やして（処理414）、判断402へ戻り、それ以降の処理を実行する。

【0161】

次に、Z値拡張処理（処理107）について説明する。

【0162】

まず、輪郭抽出基礎データ処理で抽出された各画素のZ値は、二次元マトリクス領域MMにおける平均値として算出しているため、画像全体の平均的なZ値方向に引き下げられたり、あるいは、引き上げられたりする傾向が現れる（図16参照）。

【0163】

そこで、図16に一点鎖線で示したように、Z値の画素分布が広がるように、Z値を修正する。これが、Z値拡張処理である。

【0164】

図17、18は、Z値拡張処理の一例を示している。

【0165】

ここで、*z__imageはグレースケールZ値を格納した領域をあらわすポインタ、ave__zは画像の平均Z値、max__zは画像最大Z値、min__zは画像最小Z値、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、*adjust__z__imageは修正Z値を格納する領域をあらわすポインタである。

【0166】

まず、1画面中の画素について、max__z, min__z, ave__zをそれぞれ算出する（処理501）。次いで、変数max__spanに(max__z - ave__z)の値を代入し、変数min__spanに(min__z - ave__z)の値を代入する（処理502）。

【0167】

次に、色彩深度が16ビットであるかどうかを調べて（判断503）、判断503の結果がYESになるときには、変数maxに16を代入し（処理504）、判断503の結果がNOになるときには（色彩深度が24ビットまたは32ビットの場合）、変数maxに255を代入する（処理505）。

【0168】

次いで、カウンタLnを0に初期設定し、pz_imageにz_imageを代入する（処理506）。

【0169】

そして、カウンタLnが縦幅画素数HTよりも小さいかどうかを調べる（判断507）。判断507の結果がNOになるときには、全ての画素について処理が終了した場合なので、この処理を終了する。

【0170】

判断507の結果がYESになるときには、カウンタclmnを0に初期設定し（処理508）、カウンタclmnの値が横幅画素数WDよりも小さいかどうかを調べる（判断509）。判断509の結果がNOになるときには、1ライン分の画素の処理が終了した場合なので、カウンタLnの値を1つ増やして（処理510）、判断507へ戻る。

【0171】

また、判断509の結果がYESになるときには、*pz_imageの値が画素平均値ave_zよりも大きいかどうかを調べる（判断511）。判断511の結果がYESになるときには、変数zに、 $(z - ave_z) / max_span * (max - ave_z)$ の値を代入する（処理512）。

【0172】

また、判断511の結果がNOになるときには、*pz_imageの値が画素平均値ave_zよりも小さいかどうかを調べる（判断513）。判断513の結果がYESになるときには、変数zに、 $(z - ave_z) / (min_z - ave_z) * min_span$ の値を代入する（処理514）。また、判断513の結果がNOになるとき、すなわち、*pz_imageの値が画素平均値ave_zに等しいときには、変数zに0を代入する（処理515）。

【0173】

このようにして、変数 z に値の代入が終了すると、*adjust_z_image に、*pz_image + z の値を代入し（処理 516）、注目画素について、 Z 値の修正値を格納する。

【0174】

次に、pz_image の値を 1 つ増やし、adjust_z_image の値を 1 つ増やし、c_lmn の値を 1 つ増やして（処理 517）、判断 509 へ戻り、次の画素について Z 値の修正値を算出、格納する。

【0175】

次に、参照 Z 値算出処理（処理 108）について説明する。

【0176】

まず、この参照 Z 値算出処理は、 Z 値拡張処理で算出した Z 値を修正するために参照する値を算出する処理である。

【0177】

この参照 Z 値算出処理では、図 19 に示すように、カラー画像データによりあらわされる画面 GG に含まれる 1 つ以上の物体 $OB1$ 、 $OB2$ を抽出し、今までの算出方法とは異なる手法により、それぞれの物体 $OB1$ 、 $OB2$ について、含まれる 1 つ以上の Z 値を算出し、最終的には、修正 Z 値を算出する。

【0178】

すなわち、参照 Z 値算出処理は、図 20 に示すように、カラー画像データのそれぞれの画素について所定の丸め係数で除算して丸め奥行き値を算出し、その丸め奥行き値が同一値となる画素を結ぶ等高線（図 19 の等高線 $HL1$ 、 $HL2$ 、 $HL3$ 、 $HL4$ 、 $HL5$ ）を 1 つ以上作成する等高線数値化処理（処理 701（＝等高線数値化処理ステップ））と、その作成した等高線に基づいて、カラー画像データに含まれる 1 つ以上の独立物体（図 19 の物体 $OB1$ 、 $OB2$ ）を抽出する独立物体特定処理（処理 702（＝独立物体特定処理ステップ））と、その抽出した独立物体の重心（図 19 の点 PPc ）を算出し、その重心からカラー画像データの画面 GG の下辺に垂直に下ろした垂線（図 19 の線 VL ）と、当該独立物体に含まれる 1 つ以上の等高線の交点（図 19 の点 $PP1$ 、 $PP2$ 、 $PP3$ ）

、PP4、PP5)を算出する交点算出処理(処理704(=交点算出処理ステップ))と、算出したおのこの交点について、重心のy座標、当該交点のy座標、および、あらかじめ設定されている距離情報に基づいて、第2の奥行き値を算出する新Z値算出処理(処理704(=新奥行き値算出処理ステップ))と、丸め奥行き値と新Z値の差を奥行き値修正値として算出する修正Z値算出処理(処理705(=修正奥行き値算出処理ステップ))からなる。

【0179】

また、独立物体の最も内側の等高線HL1で囲まれる部分、および、隣り合う2つの等高線で囲まれる部分を、等高面と称し、各等高面に含まれる画素には、等高線HL1に割り当てられるZ値、または、2つの等高線のうち外側の等高線に割り当てられるZ値が、それぞれ新Z値として割り当てられる。

【0180】

図21、22、23は、等高線数値化処理(処理701)の一例を示している。

【0181】

ここで、*outlineは輪郭抽出基礎データの格納領域をあらわすポインタ、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、*cordはZ値ベクトル格納用領域をあらわすポインタである。また、cordは、Vinfoという構造体のデータであり、Vinfoという構造体は、Z(グレースケール値)、npoint(整数)、および、*point(座標データ)というデータ要素を持つ。

【0182】

まず、定数maxに、255を丸め係数(後述)で除算した結果を代入し、変数Target_zに0を代入する(処理710)。次に、変数Target_zの値が定数maxよりも大きくなったかどうかを調べる(判断711)。判断711の結果がNOになるときには、この処理を終了する。

【0183】

また、判断711の結果がYESになるときには、カウンタLnを0に初期設定し(処理712)、カウンタLnの値がHTよりも小さいかどうかを調べる(

判断 713)。判断 713 の結果が NO になるときには、Target_z の値を 1 つ増やして (処理 714)、判断 711 へ戻る。

【0184】

判断 713 の結果が YES になるときには、カウンタ CLM に 0 を代入し、変数 P1 に -1 を代入し、変数 P2 に -1 を代入し、変数 fnd に 0 を代入する (処理 715)。

【0185】

カウンタ CLM の値が WD より小さいかどうかを調べ (判断 716)、判断 716 の結果が YES になるときには、*outline の値が、Target_z の値に等しいかどうかを調べる (判断 717)。判断 717 の結果が YES になるときには、変数 P1 の値が 0 以上であるかどうかを調べ (判断 718)、判断 718 の結果が NO になるときには、変数 P1 にカウンタ CLM の値を代入し (処理 719)、カウンタ CLM の値を 1 つ増やして (処理 720)、判断 716 へ戻る。また、判断 718 の結果が YES になるときには、処理 720 へ移行し、それ以降の処理を実行する。

【0186】

また、判断 717 の結果が NO になるときには、変数 P1 の値が 0 以上であるかどうかを調べ (判断 721)、判断 721 の結果が YES になるときには、変数 P2 にカウンタ CLM の値を代入し、変数 fnd の値を 1 つ増やし (処理 722)、処理 720 へ移行し、それ以降の処理を実行する。また、判断 721 の結果が NO になるときには、処理 720 へ移行し、それ以降の処理を実行する。

【0187】

また、判断 716 の結果が NO になるときには、変数 fnd の値が 0 よりも大きいかどうかを調べ (判断 723)、判断 723 の結果が NO になるときには、カウンタ Ln の値を 1 つ増やして (処理 724)、判断 713 へ戻る。

【0188】

また、判断 723 の結果が YES になるときには、当該画素位置について、Z 値が変化している画素を探索する。この Z 値変化画素探索順序は、図 22 (b) に示した順序に従う。すなわち、注目画素 P1 の直上位置にある画素の Z 値が変

化しているかどうかを調べ（判断730）、判断730の結果がNOになるときには、注目画素P1の右上位置にある画素のZ値が変化しているかどうかを調べ（判断731）、判断731の結果がNOになるときには、注目画素P1の左上位置にある画素のZ値が変化しているかどうかを調べ（判断732）、判断732の結果がNOになるときには、注目画素P1の右下位置にある画素のZ値が変化しているかどうかを調べ（判断733）、判断733の結果がNOになるときには、注目画素P1の直下位置にある画素のZ値が変化しているかどうかを調べ（判断734）、判断734の結果がNOになるときには、注目画素P1の左下位置にある画素のZ値が変化しているかどうかを調べ（判断735）、判断735の結果がNOになるときには、エラー終了する。

【0189】

また、判断730、731、732、733、734、735の結果がYESになるときには、見つけた画素の座標が既に発見済の画素であるかどうかを調べて、ベクトルを一周したかどうかを調べる（判断736）。判断736の結果がYESになるときには、この処理を終了する。

【0190】

また、判断736の結果がNOになるときには、cordのpointのxと、cordのpointのyの値に、それぞれ発見した画素の座標を保存する（処理737）。また、cordのpointの値を1つ増やし（処理738）、cordのnpointの値を1つ増やし（処理739）、注目画素P1を発見位置に移動して（処理740）、判断730へ移動し、次のベクトルについて処理を行う。

【0191】

図24、25、26は、独立物体特定処理（処理702）の一例を示している。

【0192】

ここで、*cordはZ値ベクトル格納用領域をあらわすポインタ、*cent__cordは求める中心矩形を格納するベクトル領域をあらわすポインタ、centerは求める中心点の座標データ、*side__cordは求める周縁矩

形を格納するベクトル領域をあらわすポインタである。

【0193】

まず、処理対象となるZ値ベクトルを1つ選択する(処理750)。次に、`g__min__x`に999999を代入し、`g__max__x`に0を代入し、`g__min__y`に999999を代入し、`g__max__y`に0を代入して、それらの値を初期設定する(処理751)。また、`cordp`に`cord`の値を代入する(処理752)。

【0194】

次に、`cordp`の`npoint`の値が0よりも大きいかどうかを調べ(判断753)、判断753の結果がNOになるときには、全てのベクトルについての処理が終了したかどうかを調べ(判断754)、判断754の結果がNOになるときには、処理750へ戻り、次のZ値ベクトルについての処理を行う。また、判断754の結果がYESになるときには、このときの処理を終了する。

【0195】

判断753の結果がYESになるときには、カウンタ`cnt`の値を0に初期設定し(処理755)、`min__x`に999999を代入し、`max__x`に0を代入し、`min__y`に999999を代入し、`max__y`に0を代入して、それらの値を初期設定し、変数`pnt`に`cordp`の`point`の値を代入する(処理756)。

【0196】

次いで、カウンタ`cnt`の値が`cordp`の`npoint`よりも小さいかどうかを調べ(判断757)、判断757の結果がYESになるときには、変数`pnt`の`x`の値が、`min__x`よりも小さいかどうかを調べる(判断758)。判断758の結果がYESになるときには、水平最小位置を`p1`に代入する。すなわち、`p1.x`の値に`pnt.x`の値を代入し、`p1.y`の値に`pnt.y`の値を代入し、`min__x`の値に`p1.x`の値を代入する(処理759)。また、判断758の結果がNOになるときには、処理759は実行しない。

【0197】

次いで、`pnt.x`の値が`max__x`よりも大きいかどうかを調べる(判断7

60)。判断760の結果がYESになるとときには、水平最大位置をp2に代入する。すなわち、p2.xの値にpntのxの値を代入し、p2.yの値にpntのyの値を代入し、max_xの値にp2.xの値を代入する(処理761)。また、判断760の結果がNOになるとときには、処理761は実行しない。

【0198】

次に、変数pntのyの値が、min_yよりも小さいかどうかを調べる(判断762)。判断762の結果がYESになるとときには、垂直最小位置をp3に代入する。すなわち、p3.xの値にpntのxの値を代入し、p3.yの値にpntのyの値を代入し、min_yの値にp3.yの値を代入する(処理763)。また、判断762の結果がNOになるとときには、処理763は実行しない。

【0199】

次いで、pntのyの値がmax_yよりも大きいかどうかを調べる(判断764)。判断764の結果がYESになるとときには、垂直最大位置をp4に代入する。すなわち、p4.xの値にpntのxの値を代入し、p4.yの値にpntのyの値を代入し、max_yの値にp4.yの値を代入する(処理765)。また、判断764の結果がNOになるとときには、処理765は実行しない。

【0200】

そして、カウンタcntの値を1つ増やし、pntの値を1つ増やして(処理766)、判断757へ戻る。

【0201】

また、判断757の結果がNOになるとときには、p1.xの値がg_min_xの値よりも小さいかどうかを調べ(判断767)、判断767の結果がYESになるとときには、cordpをcent_cordに複写し(処理768)、g_min_xにp1.xの値を代入する(処理769)

【0202】

次いで、点(p1.x、p3.y)と点(p2.x、p4.y)を結ぶ線分と、点(p2.x、p3.y)と点(p1.x、p4.y)を結ぶ先負との交点を求め、その交点(重心PPc)のx座標とy座標の値を、それぞれcenter

. x, center. yに保存する(処理770)。

【0203】

また、判断767の結果がNOになるとときには、処理768～770を実行しない。

【0204】

次に、p2. xがg_max_xよりも大きいかどうかを調べ(判断771)、判断771の結果がYESになるとときには、cordpをside_cordに複写し(処理772)、g_max_xにp2. xを代入する(処理773)。また、判断771の結果がNOになるとときには、処理772, 773を実行しない。

【0205】

そして、cordpの値を1つ増やし(処理774)、判断753へ戻る。

【0206】

図27は、交点算出処理(処理703)の一例を示している。

【0207】

ここで、*cordはZ値ベクトル格納用領域をあらわすポインタ、*cent_t_cordは求める中心矩形を格納するベクトル領域をあらわすポインタ、centerは求める中心点の座標データ、*side_cordは求める周縁矩形を格納するベクトル領域をあらわすポインタ、p[]は、交点の座標データを格納する配列である。

【0208】

まず、Z値ベクトルを1つ選択し(処理780)、cordpにcordを代入し、fndを0に初期設定する(処理781)。次いで、cordpのnpointの値が0よりも大きいかどうかを調べ(判断782)、判断782の結果がNOになるとときには、全ベクトルの処理が終了したかどうかを調べ(判断783)、判断783の結果がNOになるとときには、処理780へ戻り、次のZ値ベクトルについて処理を実行する。また、判断783の結果がYESになるとときには、この処理を終了する。

【0209】

判断782の結果がYESになるとときには、pntにcordpのpointの値を代入し、cntの値を0に初期設定する（処理784）。

【0210】

次に、cntが、cordpのnpointよりも小さいかどうかを調べ（判断785）、判断785の結果がYESになるとときには、pnt.xとcenter.xが等しいかどうかを調べる（判断786）。

【0211】

判断786の結果がYESになるとときには、pnt.yがcenter.yよりも大きいかどうかを調べる（判断787）。判断787の結果がYESになるとときには、中心点から垂直に下ろした垂線と等高線との交点を求める。すなわち、p[find].xにpnt.xの値を代入し、p[find].yにpnt.yの値を代入する（処理788）。

【0212】

次いで、findの値を1つ増やし（処理789）、また、データの終端を示すためにp[find].xとp[find].yにそれぞれ-1を代入する（処理790）。次に、cordpの値を1つ増やして（処理791）、判断782へ戻る。

【0213】

また、判断786の結果がNOになるとき、および、判断787の結果がNOになるとときには、cntの値を1つ増やし、pntの値を1つ増やして（処理792）、判断785へ戻る。

【0214】

また、判断785の結果がNOになるとときには、処理791へ移行する。

【0215】

図28は、新Z値算出処理（処理704）の一例を示している。

【0216】

ここで、centerは求めた中心点の座標データ、p[]は、求めた交点の座標データを格納する配列、distは、表示しているカラー画像データの画面をカメラで撮影したと設定したときのカメラから表示画面までの距離（定数）、

`focus` は上記カメラの焦点距離 (定数)、`HT` は元画像の縦幅画素数、`new_z []` は、新 Z 値を格納する配列である。

【0217】

まず、処理対象を 1 つ選択し (処理 795)、`cnt` の値を 0 に初期設定し (処理 796)、`cent_y` に `HT` から `center.y` を減じた値を代入し、`object_h` に、 $(dist / focus * center_h)$ の値を代入する (処理 797)。

【0218】

次いで、`p[cnt].x` の値が 0 以上であるかどうかを調べ (判断 798)、判断 798 の結果が YES になるとときには、`new_z[cnt]` に、 $(object_h * focus / (p[cnt].y - center.y) - dist)$ の値を代入する (処理 799)。

【0219】

そして、`cnt` の値を 1 つ増やして (処理 800)、判断 798 へ戻る。

【0220】

また、判断 798 の結果が NO になるとときには、全ての対象について処理が終了したかどうかを調べ (判断 801)、判断 801 の結果が NO になるとときには、処理 795 へ戻り、他の対象に対しての処理を行う。また、判断 801 の結果が YES になるとときには、この処理を終了する。

【0221】

図 29 は、修正 Z 値算出処理 (処理 705) の一例を示している。

【0222】

ここで、`p []` は求めた交点を格納する配列、`new_z []` は新 Z 値を格納する配列、`adjust_outline []` は修正分 Z 値を格納する配列である。

【0223】

まず、処理対象を 1 つ選択し (処理 805)、`cnt` の値に 1 を代入する (処理 806)。

【0224】

次いで、 $p[cnt].x$ の値が 0 以上であるかどうかを調べる (判断 807)。判断 807 の結果が NO になるときには、1 つの処理対象についての処理が終了したので、全ての対象に対する処理を終了したかどうかを調べる (判断 808)。判断 808 の結果が NO になるときには、処理 805 へ戻り、他の対象についての処理を行う。また、判断 808 の結果が YES になるときには、この処理を終了する。

【0225】

判断 807 の結果が YES になるときには、 $oz1$ に $(p[cnt-1].x, p[cnt-1].y-1)$ の座標の Z 値を代入し (処理 809)、 $oz2$ に $(p[cnt].x, p[cnt].y-1)$ の座標の Z 値を代入する (処理 810)。

【0226】

次いで、 $sub1$ に $(oz2-oz1)$ の値を代入し (処理 811)、 $sub2$ に $(new_z[cnt]-new_z[cnt-1])$ の値を代入する (処理 812)。

【0227】

そして、 $sub1$ と $sub2$ の符号が同一であるかどうかを調べ (判断 813)、判断 813 の結果が YES になるときには、 $adjust_outline[cnt]$ に 0 を代入し (処理 815)、判断 813 の結果が NO になるときには、 $adjust_outline[cnt]$ に $(sub2-sub1)$ の値を代入する (処理 815)。

【0228】

次いで、 cnt の値を 1 つ増やして (処理 816)、判断 807 へ戻る。

【0229】

図 30 は、Z 値修正処理 (処理 109) の一例を示している。この Z 値修正処理では、上述した等高面に含まれる全ての画素について、同一の修正 Z 値が割り当てられる。

【0230】

ここで、 $*z_image$ はグレースケール Z 値を格納している領域をあらわ

すポインタ、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、*adjust_z_imageは修正Z値を格納している領域をあらわすポインタである。

【0231】

まず、Lnに0を代入し（処理901）、Lnの値がHTより小さいかどうかを調べる（判断902）。判断902の結果がNOになるときには、この処理を終了する。

【0232】

判断902の結果がYESになるときには、clmnに0を代入して（処理903）、clmnがWDよりも小さいかどうかを調べる（判断904）。判断904の結果がYESになるときには、そのときの注目画素に対応するadjust_z_imageを取得し（処理905）、z_imageを、z_imageとadjust_z_imageを加算した値で更新する（処理906）。

【0233】

次いで、clmnの値を1つ増やし（処理907）、判断904へ戻る。判断904の結果がNOになるときには、Lnの値を1つ増やして（処理908）、判断902へ戻り、それ以降の処理を行う。

【0234】

図31は、立体変換出力処理（処理110）の一例を示している。

【0235】

ここで、*srcは元画像の格納領域をあらわすポインタ、*outinfoは輪郭抽出基礎データを格納した領域をあらわすポインタ、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、*image3dは立体視画像データを格納する領域をあらわすポインタである。

【0236】

まず、LNを0に初期設定し（処理1001）、LNがHTより小さいかどうかを調べる（判断1002）。判断1002の結果がNOになるときには、この処理を終了する。また、1002の結果がYESになるときには、CNを0に初期設定する（処理1003）、。

【0237】

次いで、CNがWDよりも小さいかどうかを調べ（判断1004）、判断1004の結果がYESになるとときには、そのときのoutinfoに対応した適用Z値（ZZ）を算出する適用Z値算出処理（処理1005）を行い、続いて、適用Z値に基づいて視差値（SS）を算出する視差値算出処理（処理1006）を行う。

【0238】

このようにして、視差SSが算出されると、LNの値が偶数であるかどうかを調べ（判断1007）、判断1007の結果がYESになるとときには、MVに（CN+SS）の値を代入し（処理1008）、また、判断1007の結果がNOになるとときには、MVに（CN-SS）の値を代入する（処理1009）。

【0239】

次いで、元画像データの（src+MV）の位置の画素のデータをimage3dへ格納する（処理1010）。

【0240】

次に、image3dの値を1つ増やし、outinfoの値を1つ増やし、CNの値を1つ増やして（処理1011）、判断1004へ戻る。

【0241】

また、判断1004の結果がNOになるとときには、srcを次の行頭へ進め（処理1012）、LNの値を1つ増やして（処理1013）、判断1002へ戻り、それ以降の処理を行う。

【0242】

図32は、適用Z値算出処理（処理1005）の一例を示している。

【0243】

まず、256をオブジェクト深度（上述した設定ダイアログパネルの入力窓IB2に入力された値）で除算して、その値を丸め係数MKに代入し（処理1020）、z_imageの値を丸め係数MKで除算した結果を、ZZに代入する（処理1021）。なお、ここで算出した丸め係数MKは、処理710で用いる丸め係数と同じものである。すなわち、処理710における丸め係数は、この丸め

係数MKを用いる。

【0244】

ここで、ZZの値が上限基準値（上述した設定ダイアログパネルの入力窓IB3に入力された上限の値）よりも大きいかどうかを調べ（判断1022）、判断1022の結果がYESになるとときには、ZZに上限基準値を代入する（処理1023）。また、判断1022の結果がNOになるとときには、処理1023を実行しない。

【0245】

次いで、ZZの値が下限基準値（上述した設定ダイアログパネルの入力窓IB4に入力された下限の値）よりも小さいかどうかを調べ（判断1024）、判断1024の結果がYESになるとときには、ZZに下限基準値を代入する（処理1025）。また、判断1024の結果がNOになるとときには、処理1025を実行しない。

【0246】

図33は、視差値算出処理（処理1006）の一例を示している。

【0247】

まず、処理105のZ値反転判定処理において、Z値を反転する旨が判定出力されているかどうかを調べる（判断1030）。

【0248】

判断1030の結果がYESになるとときには、ZZに $(256 - ZZ)$ の値を代入することで、ZZの値を修正する（処理1031）。次いで、ZZに奥行深度（上述した設定ダイアログパネルの入力窓IB1に入力された値）を加算して、その結果を視差値SSに代入して（処理1032）、この処理を終了する。

【0249】

また、判断1030の結果がNOになるとときには、処理1032へ移行し、Zに奥行深度を加算した値を視差値SSに代入して、この処理を終了する。

【0250】

このようにして、本実施例では、Z値算出処理により、Z値の元になるグレースケール値として、人間の網膜に存在する赤錐体、緑錐体、および、青錐体の数

の分布に基づき、立体視画像を作成する際に特に適する重み付けの値を与えたRデータの30%と、Gデータの59%と、Bデータの11%の総和を取った値を用いているので、人間がより自然と感じる奥行き値を算出することができる。なお、このRGBデータに対する重み付けの値の態様は、それ以外の組み合わせを採用することもできる。

【0251】

また、Z値反転判断処理により、白い部分を遠くにするか、あるいは、黒い部分を遠くにするかを判断しており、それにより、不自然な立体感を除去し、ごく自然な立体感を実現することができる。

【0252】

また、参照Z値算出処理により、画像データに含まれる1つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのおのの奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、さらに、その等高面の奥行き値を、あらかじめ設定した距離条件に従って補正しているので、奥行き値の順序性を正しいものとして修正することができ、それにより、不自然な立体感を除去し、ごく自然な立体感を実現することができる。

【0253】

また、輪郭抽出基礎データ処理により、二次元マトリクス領域MMの範囲で平均値を算出して、注目画素 P_x のZ値として割り当てるようにしているので、極端にZ値が変化している部分のZ値の変化値がなめらかになり、その結果、画素のクロス現象を抑制することができ、画像の乱れが抑制される。また、物体が重なっている部分では、その物体の境界を構成する輪郭が際だつこととなり、結果的に、おのおのの物体の独立性が保持され、その部分での物体の区別が強調される。

【0254】

また、Z値が高い画素やZ値が低い画素の数は、画像データ全体の画素数に比べて非常に少ない数であるから、上限基準値や下限基準値の処理で特定値を超えるZ値を一定範囲内に収まるように再配置しているので、一定範囲内に納めたZ値を画素クロスが発生しない限度一杯に拡張することで、立体感を損なわずに画

像のゆがみを抑えることができることとなった。

【0255】

また、ユーザが立体視画像表示アプリケーションの表示画面で適宜に設定した入力値、および、設定ダイアログパネルで適宜に入力した入力値に従って、立体視画像データの視差値が算出されるので、ユーザは、所望する態様の立体視画像を適宜に表示させることができる。

【0256】

また、上述した処理は、動画の1フレームについて行われるので、ユーザが上述した入力値を適宜に変更すると、その変更を反映した立体視画像または非立体視画像が、次のフレームから表示されるので、非常に便利であり、使い勝手が良好である。

【0257】

ここで、本実施例では、立体視画像表示アプリケーションの表示画面の項目M I 8をクリック操作する毎に、表示モードが非立体視モードから立体視モードへと切り替わり、その切り替わりのたびに、上述した処理が開始され、その結果、次のフレームからは、立体視モードの動画がメイン画面MDへと表示される。

【0258】

ところで、上述した実施例では、立体視画像の表示方式として、C R T表示装置11の表示画面の偶数ラインに左目画像を表示するとともに、奇数ラインに右目画像を表示するようにしたいいわゆるO d d - e v e n方式のものを採用しているが、本発明は、C R T表示装置11の表示画面の偶数ラインに右目画像を表示するとともに、奇数ラインに左目画像を表示するようにしたO d d - e v e n方式のものにも適用できる。また、本発明はそれ以外の表示方式のものについても、同様にして適用することができる。

【0259】

例えば、図34(a)に示したように、左目用画像データの偶数ラインを抽出してフレームバッファの上画像部分に格納するとともに、右目用画像データの奇数ラインを抽出してフレームバッファの下画像部分に格納する上下方式にも適用できる。この表示方式は、立体視プロジェクタ等で採用されている。また、右目

用画像データの偶数ラインを抽出してフレームバッファの上画像部分に格納するとともに、左目用画像データの奇数ラインを抽出してフレームバッファの下画像部分に格納する上下方式にも適用できる。

【0260】

また、同図（b）に示したように、左目用画像データの偶数番目ピクセルを抽出してフレームバッファの左側画像部分に格納するとともに、右目用画像データの奇数番目ピクセルを抽出してフレームバッファの右側画像部分に格納する左右方式にも適用できる。この表示方式は、立体裸眼CRT等に採用されている。また、右目用画像データの偶数番目ピクセルを抽出してフレームバッファの左側画像部分に格納するとともに、左目用画像データの奇数番目ピクセルを抽出してフレームバッファの右側画像部分に格納する左右方式にも適用できる。

【0261】

また、図35（a）に示したように、左目用画像データを偶数番目フレームのフレームバッファに格納するとともに、右目用画像データの奇数番目のフレームバッファに格納する2フレーム方式にも適用できる。この表示方式は、立体視プロジェクタ等で採用されている。また、右目用画像データを偶数番目フレームのフレームバッファに格納するとともに、左目用画像データの奇数番目のフレームバッファに格納する2フレーム方式にも適用できる。

【0262】

また、同図（b）に示したように、左目用画像データを左目用画像データファイルに格納するとともに、右目用画像データを右目用画像データファイルに格納するOdd-evenファイル分離方式にも適用できる。この表示方式は、立体視プロジェクタ等で採用されている。

【0263】

図36は、立体変換出力処理の他の例を示している。この場合には、左目用画像データと右目用画像データを、それぞれLimage3dとRimage3dで示される左目用画像データ領域と右目用画像データ領域に格納している。

【0264】

まず、LNを0に初期設定し（処理1051）、LNがHTより小さいかどうか

かを調べる（判断1052）。判断1052の結果がNOになるときには、この処理を終了する。また、1052の結果がYESになるときには、CNを0に初期設定する（処理1053）、。

【0265】

次いで、CNがWDよりも小さいかどうかを調べ（判断1054）、判断1054の結果がYESになるときには、そのときのoutinfoに対応した適用Z値（ZZ）を算出する適用Z値算出処理（処理1055；図32の処理）を行い、続いて、適用Z値に基づいて視差値（SS）を算出する視差値算出処理（処理1056；図33の処理）を行う。

【0266】

このようにして、視差値SSが算出されると、MV1に（CN+SS）の値を代入するとともに、MV2に（CN-SS）の値を代入し（処理1057）、次いで、元画像データの（src+MV1）の位置の画素のデータをLimage3dへ格納し（処理1058）、元画像データの（src+MV2）の位置の画素のデータをRimage3dへ格納する（処理1059）。

【0267】

次に、Limage3dの値を1つ増やし、Rimage3dの値を1つ増やし、outinfoの値を1つ増やし、CNの値を1つ増やして（処理1060）、判断1054へ戻る。

【0268】

また、判断1054の結果がNOになるときには、srcを次の行頭へ進め（処理1061）、LNの値を1つ増やして（処理1062）、判断1052へ戻り、それ以降の処理を行う。

【0269】

ところで、図19で示したような画面GGに含まれる独立物体について、例えば、図37に示すように、立体感を意図的に強調するような効果を与えることもできる。

【0270】

図37の場合、独立物体OBaについて、それに含まれる全ての等高面（同一

【0 2 7 1】

【0 2 7 2】

【0 2 7 3】

【0 2 7 4】

【0 2 7 5】

【0 2 7 6】

出証特 2 0 0 3 - 3 0 7 8 3 9 7

スケール値との関係に基づいて、そのフレームの画面の明るさの傾向が、画面の下から上方向へ増大しているか、あるいは、中央部から周辺部へ向かって増大しているか、もしくは、画面の上から下方向へ増大しているか、あるいは、中央部から周辺部へ向かって減少しているかを判断し、前者の場合には、グレースケール値の変化傾向と奥行き値の変化傾向が同一であると判断する一方、後者の場合には、グレースケール値の変化傾向に対して奥行き値の変化傾向が反転していると判断する Z 値反転判断処理（処理 1105（＝奥行き値反転判断ステップ））を行う。

【0277】

次いで、カラー画像データのそれぞれの画素について、当該注目画素（処理対象となっている画素；以下同じ）を含む所定サイズの二次元マトリクスに含まれるグレースケール値の平均値を、当該画素の奥行き値として割り当てる輪郭抽出基礎データ処理（処理 1106（＝奥行き値算出ステップ））を行う。

【0278】

次に、カラー画像データのそれぞれの画素について処理 1106 で算出した奥行き値の画面全体の平均値と、最大値と、最小値に基づいて、それぞれの画素について、奥行き値を修正する Z 値拡張処理（処理 1107（＝第 1 の奥行き値修正ステップ））を行う。

【0279】

次に、カラー画像データに含まれる 1 つ以上の独立物体を抽出し、その抽出した独立物体に含まれる等高線（後述）について、新たに奥行き値を算出し、その新たに算出した奥行き値に基づいて、奥行き値修正値を算出する参照 Z 値算出処理（処理 1108（＝参照奥行き値算出ステップ））を行う。

【0280】

次いで、おのおのの画素について、処理 1108 で算出した奥行き値に、上記参照奥行き値算出ステップで算出した奥行き値修正値を加算して奥行き値を修正する Z 値修正処理（処理 1109（＝第 2 の奥行き値修正ステップ））を行う。

【0281】

次に、独立物体の立体感を強調するための独立物体立体感強調処理（処理 11

10) を行い、それぞれの Z 値をさらに修正する。

【0282】

次に、おのこの画素について、それまでに算出して得られたデータに基づいて、左目用画像データおよび右目用画像データを作成して出力する立体変換出力処理（処理1111（＝適用奥行き値算出ステップ、視差値算出ステップ、および、画像データ作成ステップ））を行う。

【0283】

ここで、処理1104, 1105, 1106, 1107, 1108, 1109, 1111は、それぞれ前述した処理104, 105, 106, 107, 108, 109, 110と同一であるので、その説明を省略する。

【0284】

図39は、独立物体立体感強調処理（処理1110）の一例を示している。

【0285】

ここで、*z__imageはグレースケールZ値の格納領域をあらわすポインタ、WDは元画像の横幅画素数、HTは元画像の縦幅画素数、*side__cordは強調する物体の周縁矩形、offset__zは強調するZ値である。

【0286】

まず、Lnの値を0に設定し（処理1120）、LnがHTよりも小さいかどうかを調べる（判断1121）。判断1121の結果がNOになるとときには、この処理を終了する。また、判断1121の結果がYESになるとときには、clmnの値を0に設定する（処理1122）。

【0287】

次に、clmnがWDより小さいかどうかを調べ（判断1123）、判断1123の結果がYESになるとときには、座標（clmn、Ln）から画像上辺までの半直線を引き（処理1124）、処理1124で引いた半直線とside__cordとの交点数を求める（処理1125）。

【0288】

そして、処理1125で求めた交点数が奇数であるかどうかを調べる（判断1126）。判断1126の結果がYESになるとときには、座標（clmn、Ln

) は、周知の偶奇規則より周縁矩形の内部であると判定し (処理 1127)、 p_z_image に $(z_image + L_n * WD + c_l_m_n)$ を代入し (処理 1128)、 $*p_z_image$ に $(*p_z_image + offset_z)$ の値を代入する (処理 1129)。また、判断 1126 の結果が NO になるときには、処理 1127 ~ 1129 を実行しない。

【0289】

次いで、 L_n の値を 1 つ増やして、判断 1123 へ戻る。また、判断 1123 の結果が NO になるときには、 L_n の値を 1 つ増やして判断 1121 へ戻る。

【0290】

ところで、上述した各実施例では、処理対象とする画像データを、二次元カラー画像データとしているが、本発明は、白黒映画の動画データのように、二次元白黒多階調画像データについても同様にして適用することができる。

【0291】

また、上述した各実施例では、処理対象とする画像データを動画データの 1 フレーム分のデータとしているが、本発明は、静止画データを処理対象とすることもできる。

【0292】

【発明の効果】

以上説明したように、本発明によれば、各画素の画像データが所定ビット深さを持つ二次元画像データのそれぞれの画素について、画像データ値に基づいて奥行き値を算出し、その奥行き値に基づいて立体視画像データの視差値を算出する立体視画像データ算出方法において、上記二次元画像データの 1 画面分の奥行き値の平均値と、上記二次元画像データの複数サンプリング領域の奥行き値との関係に基づいて、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が同一であるか、上記奥行き値の変化傾向と画面上の所定の基準点からの距離の変化傾向が反転していると判断する奥行き値反転判断ステップと、上記二次元カラー画像データのそれぞれの画素について、1 画面分の奥行き値の平均値と、奥行き値の最大値と、奥行き値の最小値に基づいて、それぞれの画素について、奥行き値を、その分布が拡張する態様に修正した修正奥行き値を算出する第

1 の奥行き値修正ステップと、上記二次元画像データの 1 画面に含まれる 1 つ以上の独立物体を抽出し、その独立物体の周縁部を結ぶ等高線をおのこの奥行き値について形成するとともに、同一奥行き値を有する等高面を決定し、その等高面の奥行き値を、あらかじめ設定した距離条件に従って算出する参照奥行き値算出ステップと、おのこの画素について、当該画素が含まれる上記等高面での参照奥行き値の変化態様が、上記修正奥行き値の変化態様と異なっている場合には、修正奥行き値の順序が保存される態様にその修正奥行き値を修正し、第 2 の修正奥行き値を算出する第 2 の奥行き値修正ステップと、おのこの画素について、上記第 2 の修正奥行き値に基づき所定の丸め係数を用いて丸め奥行き値を算出し、その丸め奥行き値をあらかじめ設定された上限値と下限値で定められる範囲の値に制限して第 2 の丸め奥行き値を算出する適用奥行き値算出ステップと、処理対象となる画素について、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が同一であると判断されている場合には、当該画素について算出された上記第 2 の丸め奥行き値に所定の奥行き深度を加えた値を視差値として算出する一方、上記奥行き値反転判断ステップで上記奥行き値の変化傾向と距離の変化傾向が反転していると判断されている場合には、当該画素について算出された上記第 2 の丸め奥行き値を、奥行き値の取りうる最大値より減じた値に、上記奥行き深度を加えた値を、当該画素の視差値として算出する視差値算出ステップと、処理対象となる画素について、上記視差値算出ステップで算出された視差値、および、上記二次元画像データにおける原座標に基づいて、当該画素の画像データを、右目用画像データまたは左目用画像データのそれぞれの記憶領域の対応する座標領域に保存する画像データ作成ステップより立体視画像データを作成しているので、自然な立体感を得ることができるという効果を得る。

【図面の簡単な説明】

【図 1】

本発明の一実施例にかかる立体視用画像データ作成装置の一例を示したブロック図。

【図 2】

立体視画像表示アプリケーションの表示画面の一例を示した概略図。

【図 3】

設定ダイアログパネルの一例を示した概略図。

【図 4】

本発明の一実施例にかかる立体視表示機構を説明するための概略図。

【図 5】

本発明の一実施例にかかる立体視表示機構を説明するための概略図。

【図 6】

立体視画像表示アプリケーションにおいて、立体視表示モードが選択された場合に行われる処理の一例を示したフローチャート。

【図 7】

カラー画像データの 1 画素分のデータ構造を説明するための概略図。

【図 8】

Z 値算出処理（処理 1 0 4）の一例を示したフローチャート。

【図 9】

Z 値算出処理（処理 1 0 4）の一例を示したフローチャート（続き）。

【図 1 0】

Z 値算出処理（処理 1 0 4）の一例を示したフローチャート（続き）。

【図 1 1】

Z 値判定反転処理で適用するセンサの配置例を示した概略図。

【図 1 2】

Z 値反転判定処理の一例を示したフローチャート。

【図 1 3】

Z 値反転判定処理の一例を示したフローチャート（続き）。

【図 1 4】

輪郭抽出基礎データ処理において画素値を算出するための二次元マトリクス領域 MM を説明するための概略図。

【図 1 5】

輪郭抽出基礎データ処理（処理 1 0 6）の一例を示したフローチャート。

【図 1 6】

Z 値拡張処理を説明するためのグラフ図。

【図 17】

Z 値拡張処理の一例を示したフローチャート。

【図 18】

Z 値拡張処理の一例を示したフローチャート（続き）。

【図 19】

参照 Z 値算出処理で適用するモデルを説明するための概略図。

【図 20】

参照 Z 値算出処理の一例を示したフローチャート。

【図 21】

等高線数値化処理（処理 701）の一例を示したフローチャート。

【図 22】

等高線数値化処理（処理 701）の一例を示したフローチャート（続き）。

【図 23】

等高線数値化処理（処理 701）の一例を示したフローチャート（続き）。

【図 24】

独立物体特定処理（処理 702）の一例を示したフローチャート。

【図 25】

独立物体特定処理（処理 702）の一例を示したフローチャート（続き）。

【図 26】

独立物体特定処理（処理 702）の一例を示したフローチャート（続き）。

【図 27】

交点算出処理（処理 703）の一例を示したフローチャート。

【図 28】

新 Z 値算出処理（処理 704）の一例を示したフローチャート。

【図 29】

修正 Z 値算出処理（処理 705）の一例を示したフローチャート。

【図 30】

Z 値修正処理（処理 109）の一例を示したフローチャート。

【図 3 1】

立体変換出力処理（処理 1 1 0）の一例を示したフローチャート。

【図 3 2】

適用 Z 値算出処理（処理 1 0 0 5）の一例を示したフローチャート。

【図 3 3】

視差値算出処理（処理 1 0 0 6）の一例を示したフローチャート。

【図 3 4】

立体視画像の表示方式の他の例を説明するための概略図。

【図 3 5】

立体視画像の表示方式のさらに他の例を説明するための概略図。

【図 3 6】

立体変換出力処理の他の例を示したフローチャート。

【図 3 7】

独立物体に対する立体感の強調について説明するための概略図。

【図 3 8】

立体視表示モードが選択された場合に行われる処理の他の例を示したフローチャート。

【図 3 9】

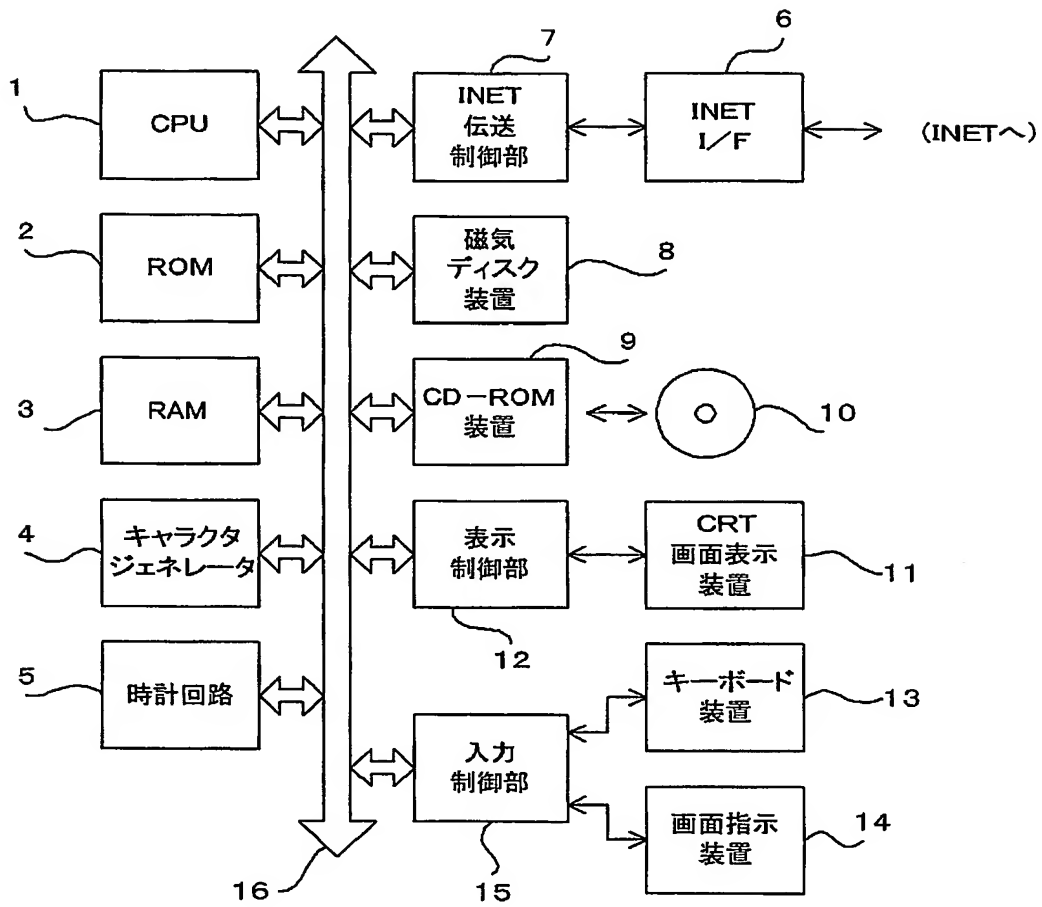
独立物体立体感強調処理（処理 1 1 1 0）の一例を示したフローチャート。

【符号の説明】

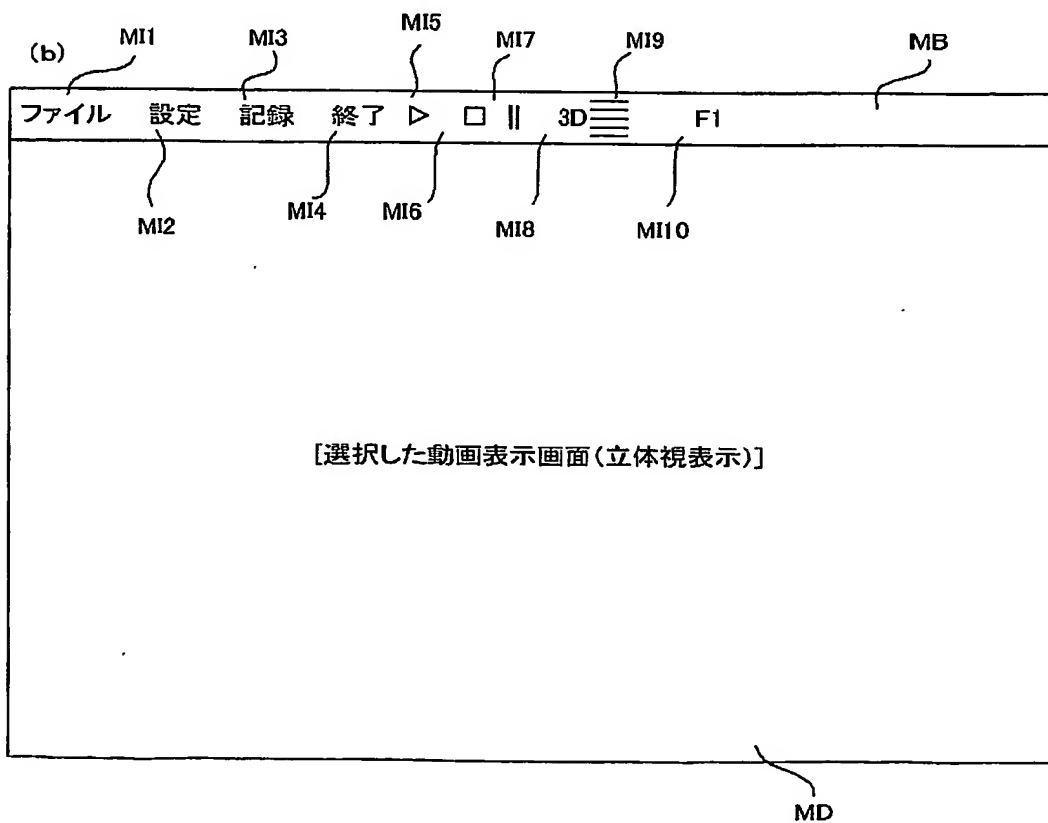
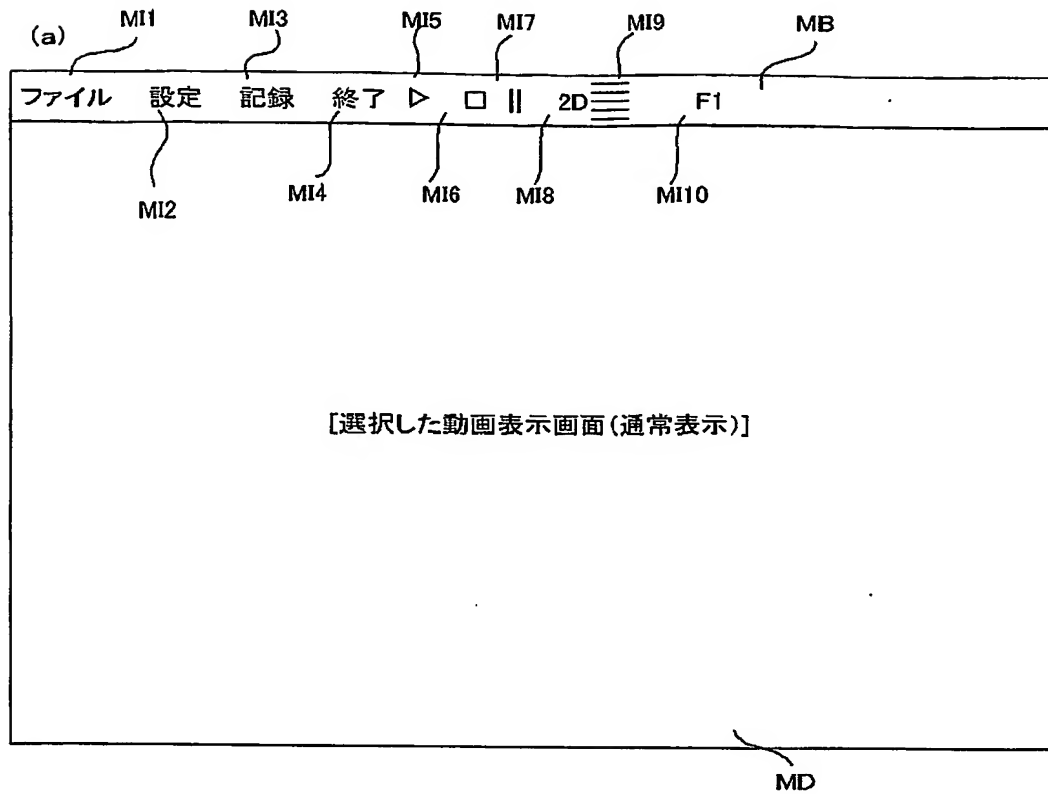
- 1 CPU（中央処理装置）
- 2 ROM（リード・オンリ・メモリ）
- 3 RAM（ランダム・アクセス・メモリ）

【書類名】 図面

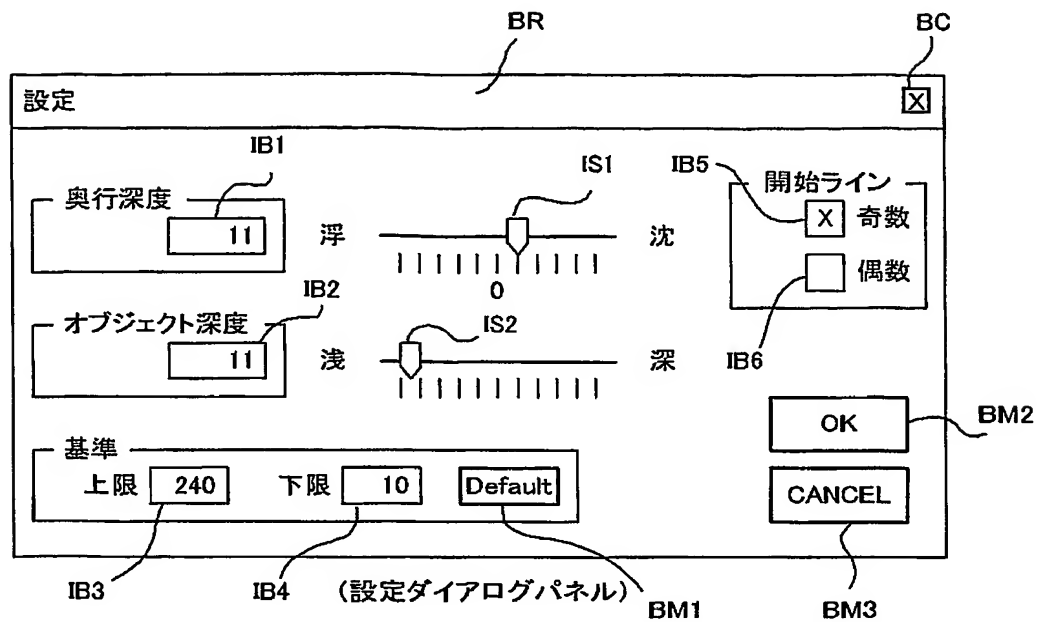
【図 1】



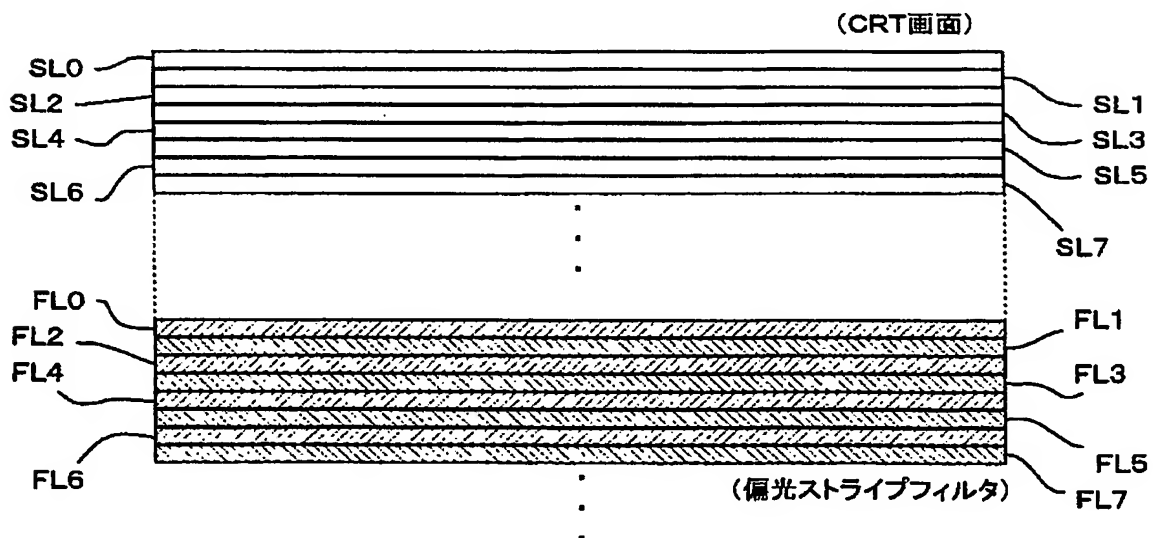
【図 2】



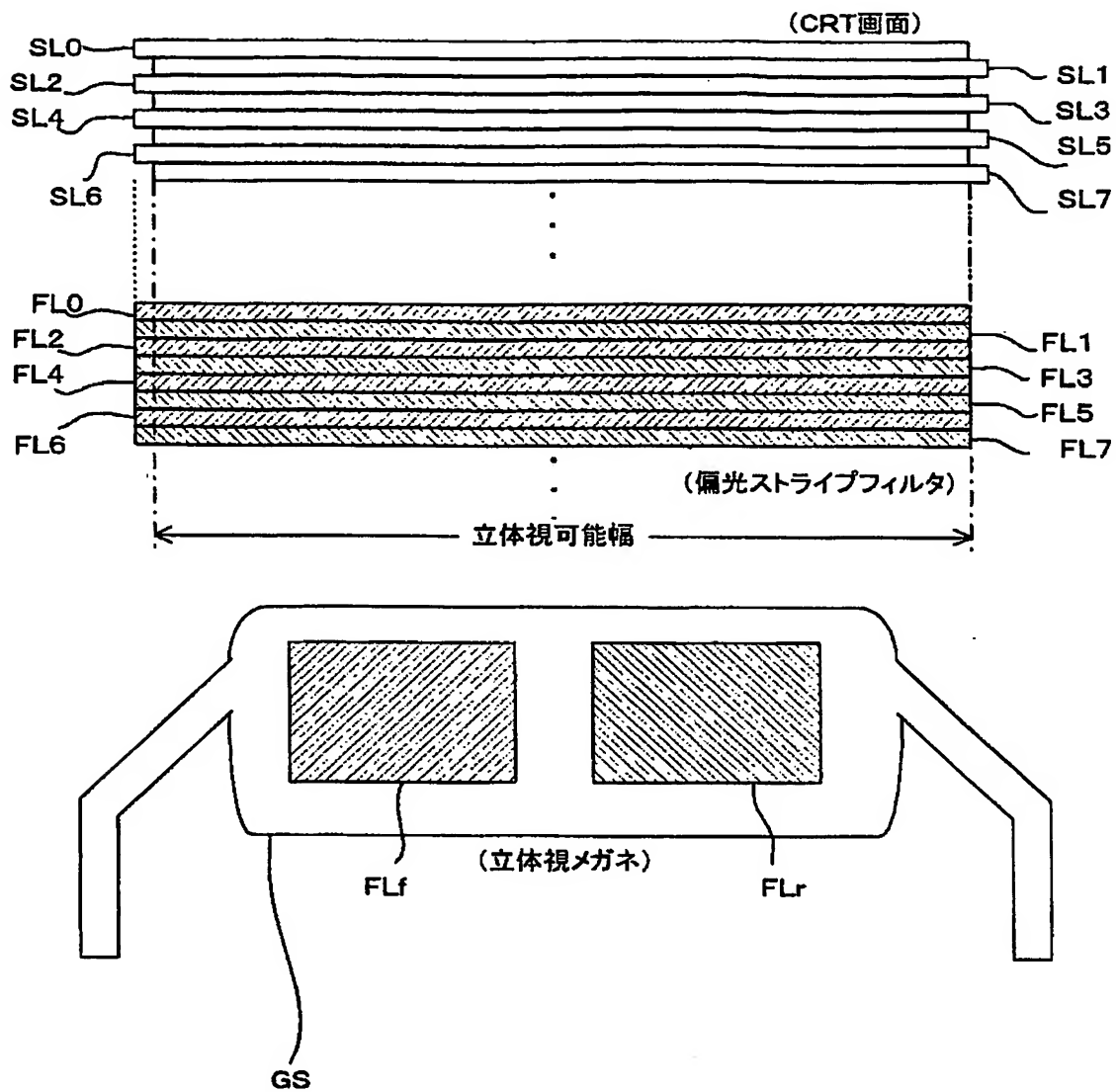
【図 3】



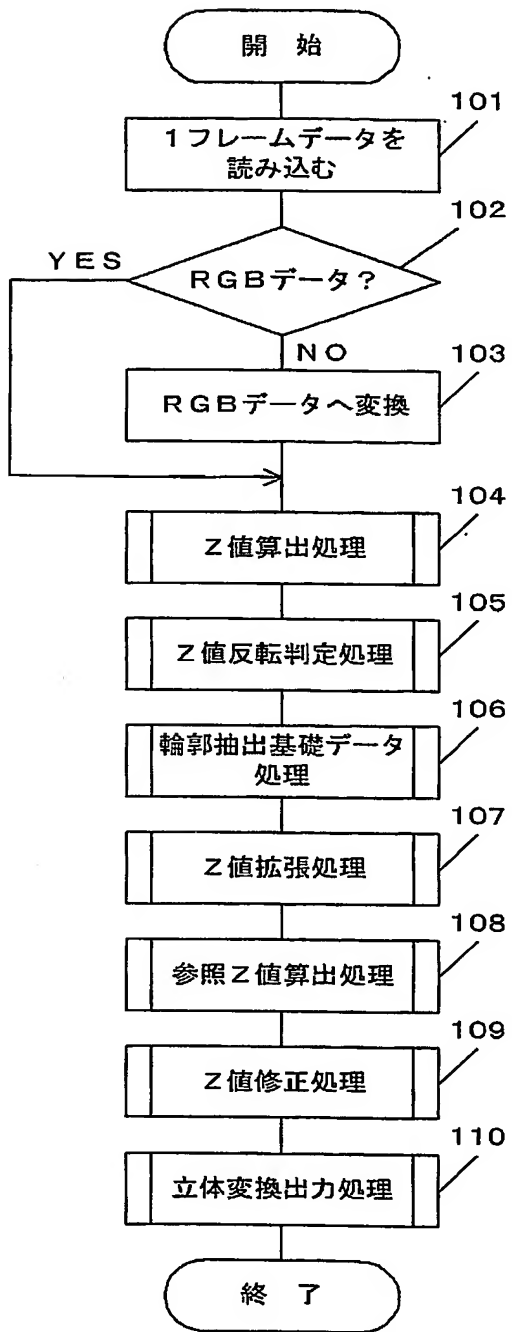
【図 4】



【図 5】



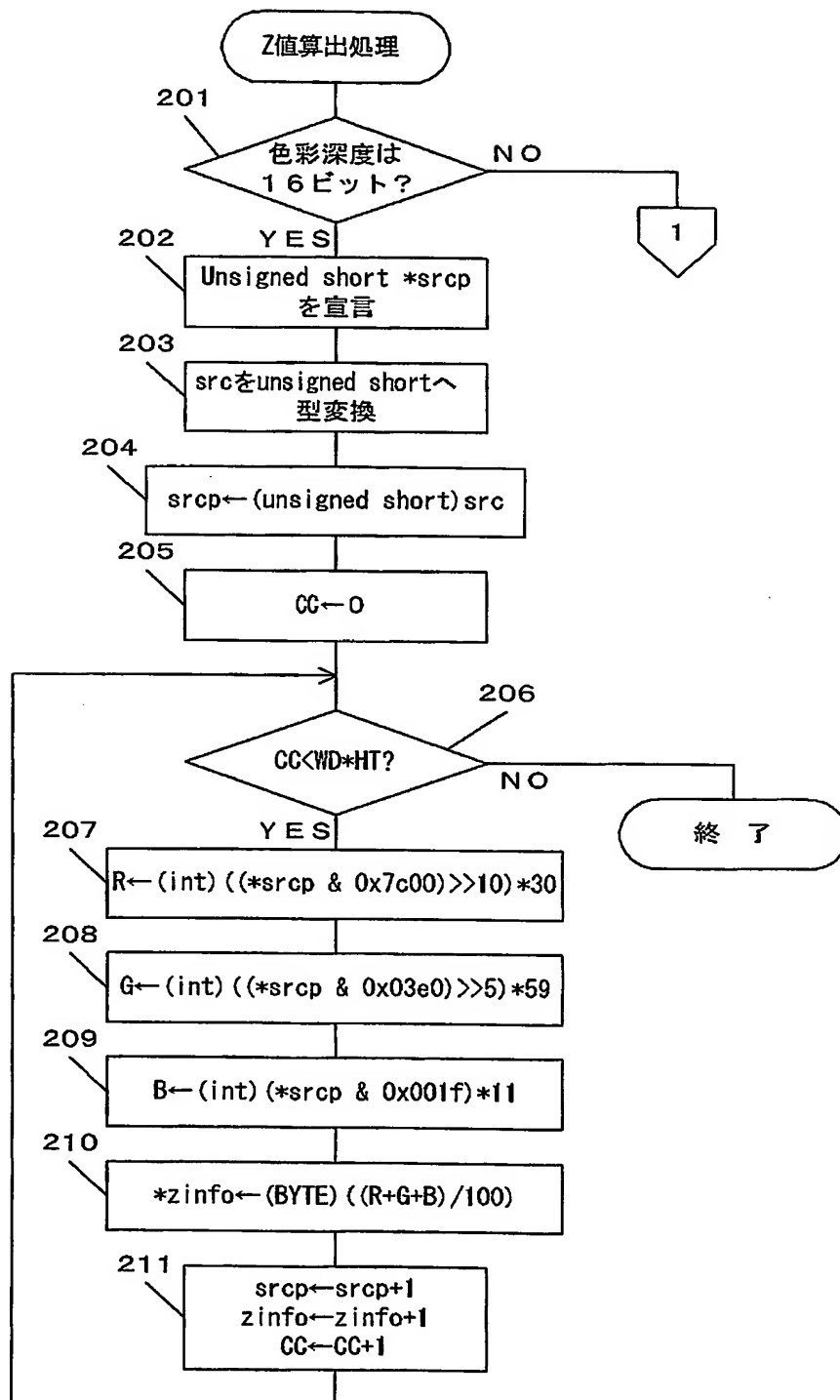
【図 6】



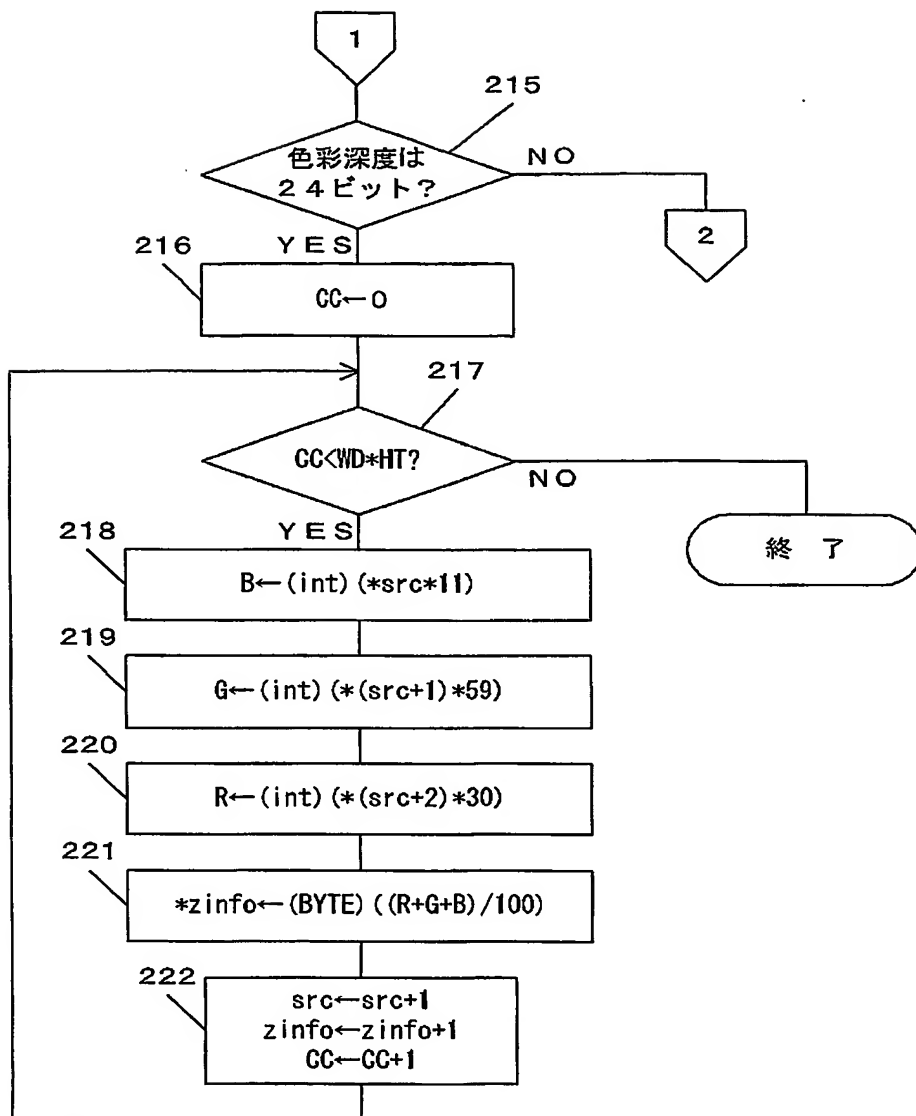
【図 7】



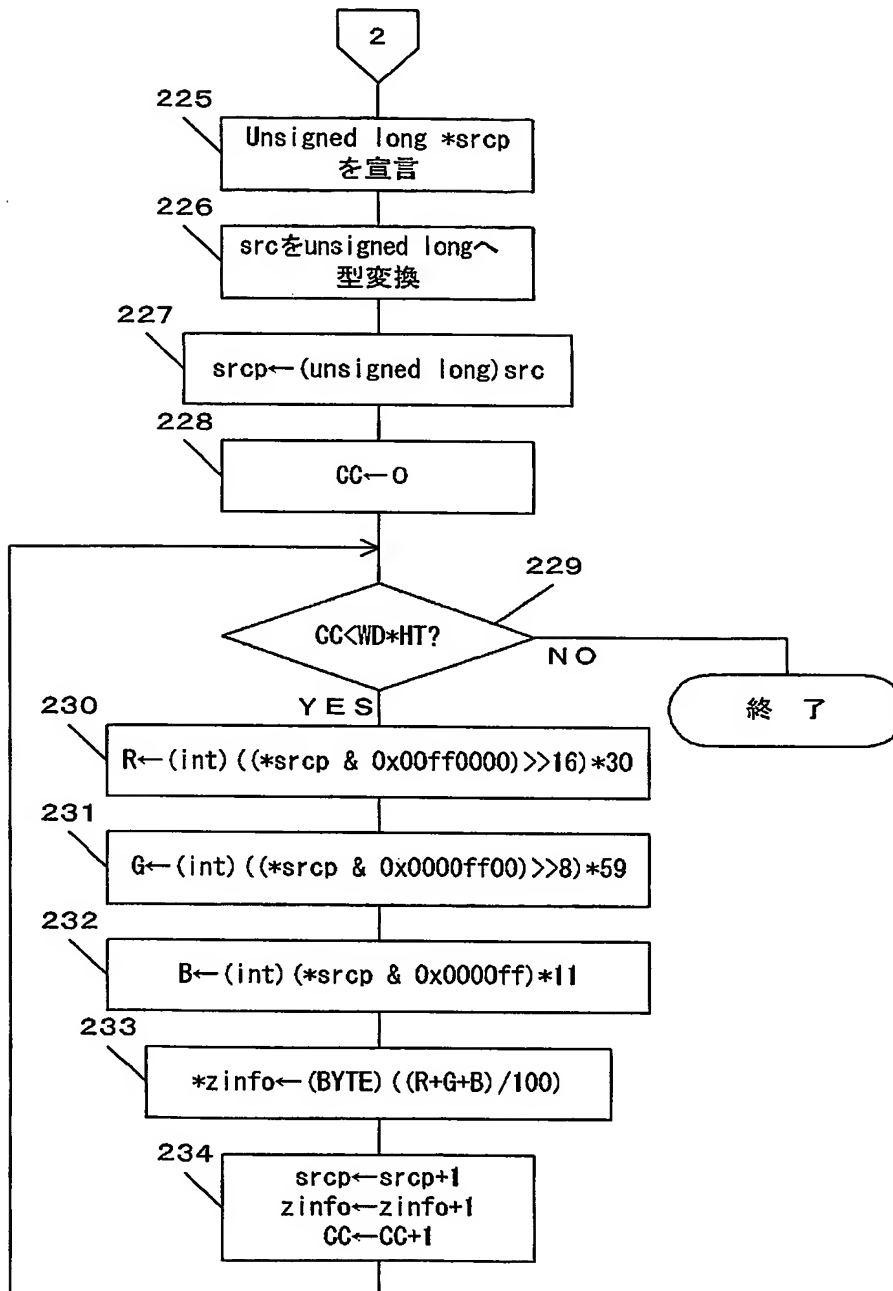
【図 8】



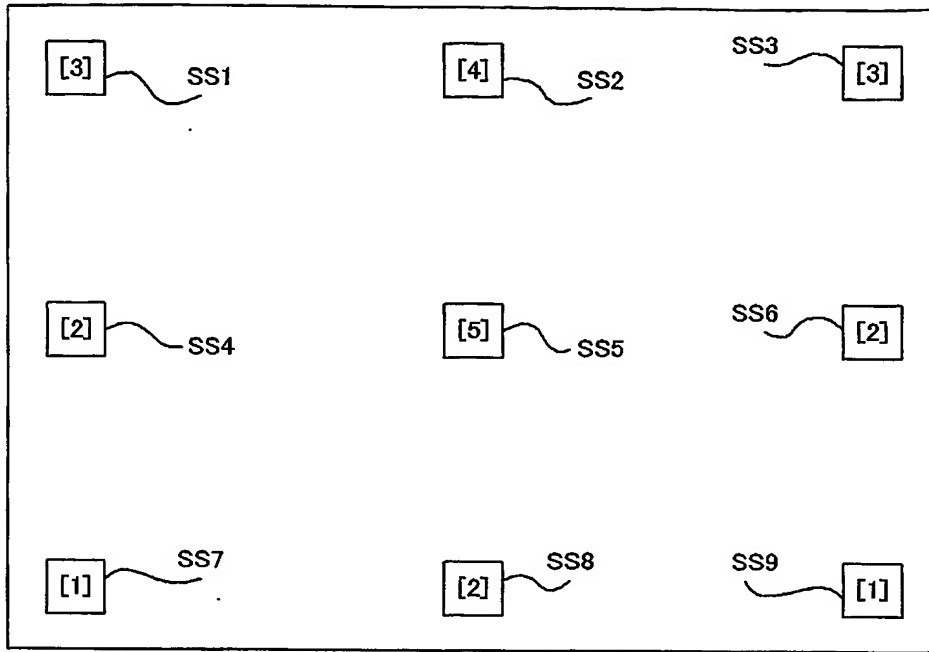
【図 9】



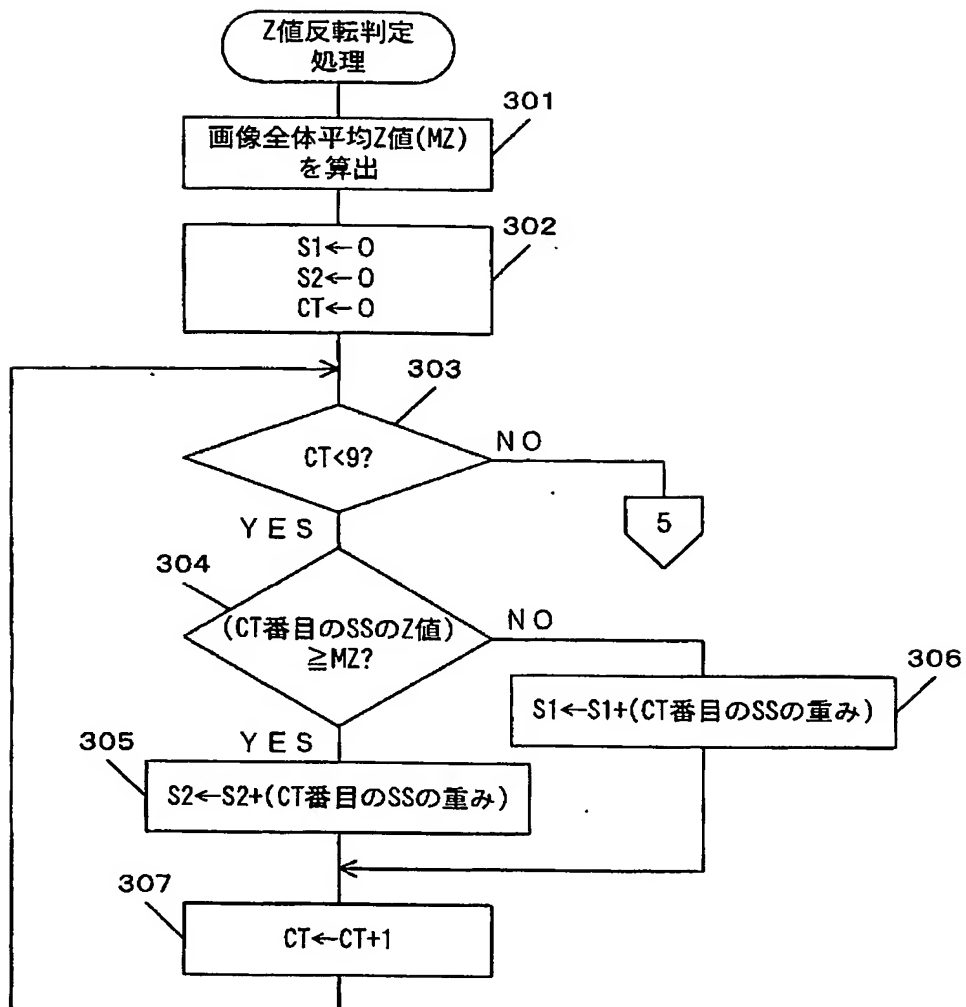
【図 10】



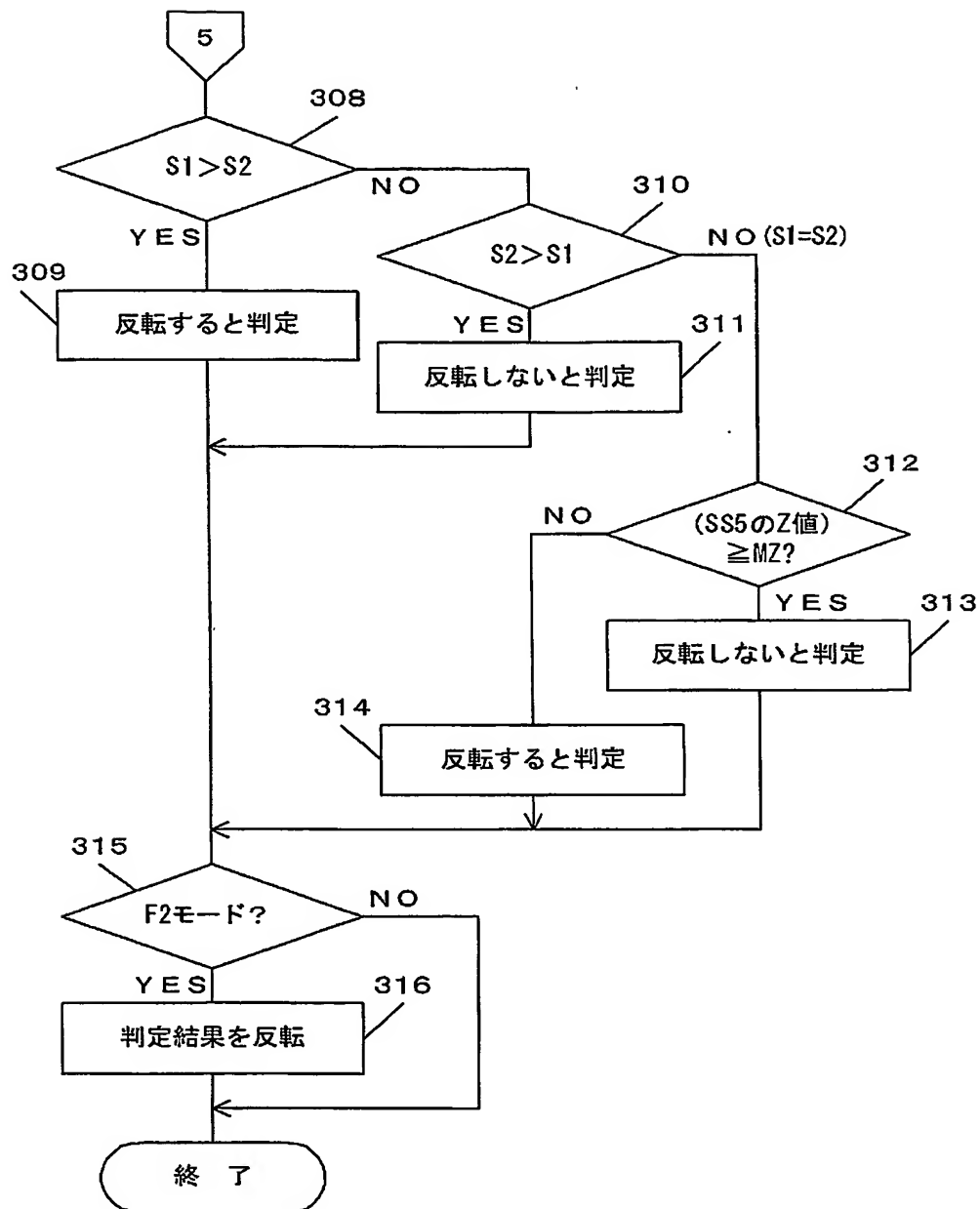
【図 11】



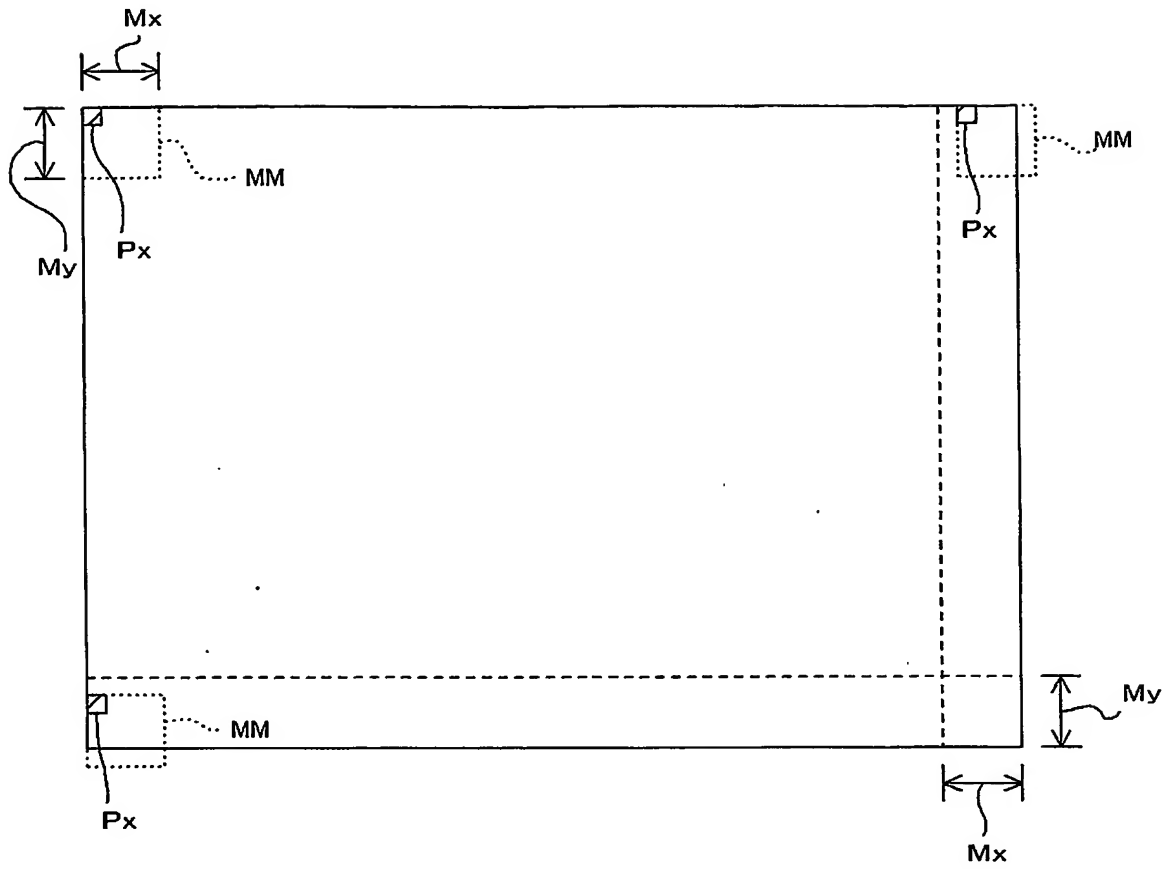
【図 12】



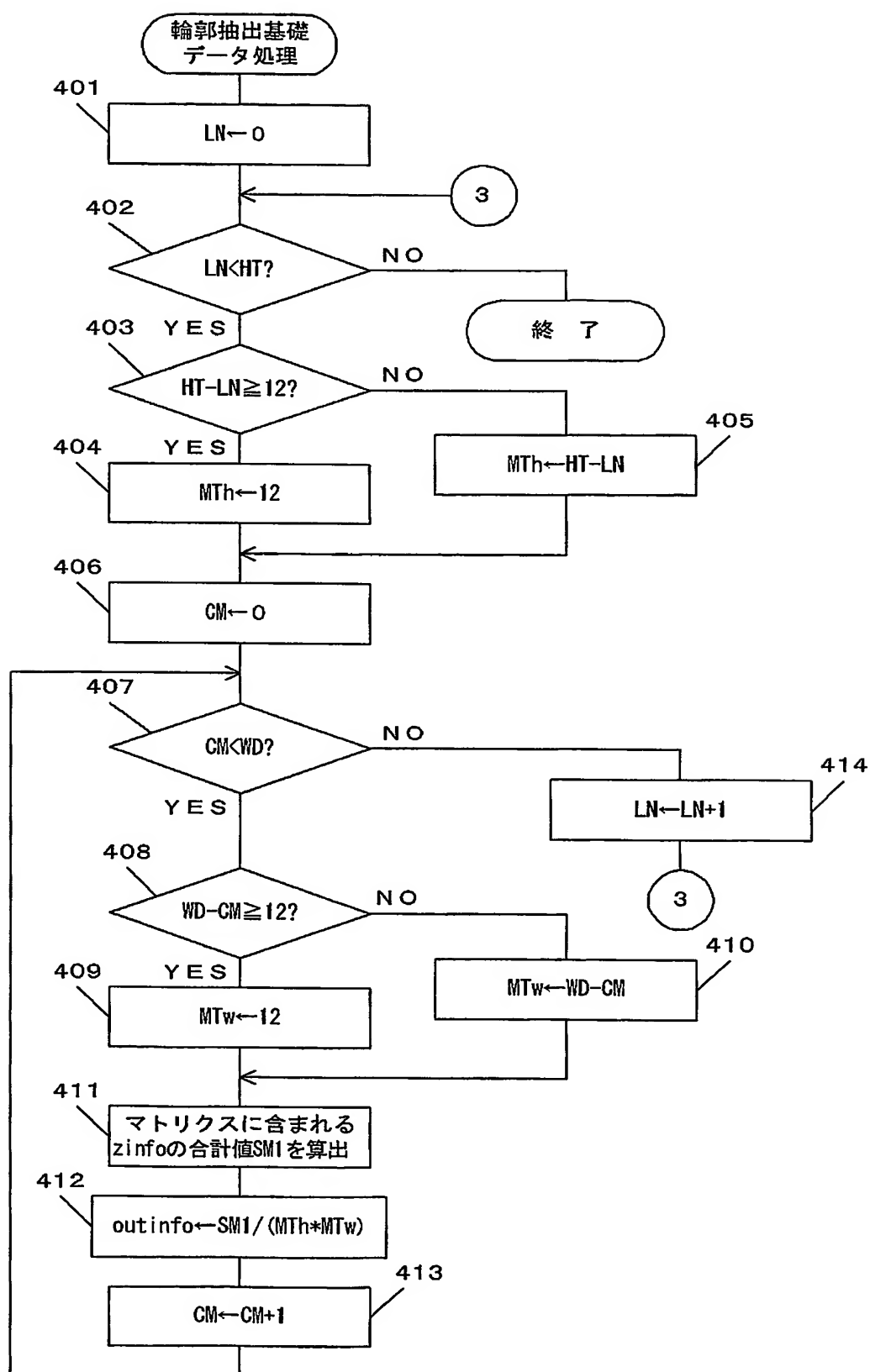
【図 13】



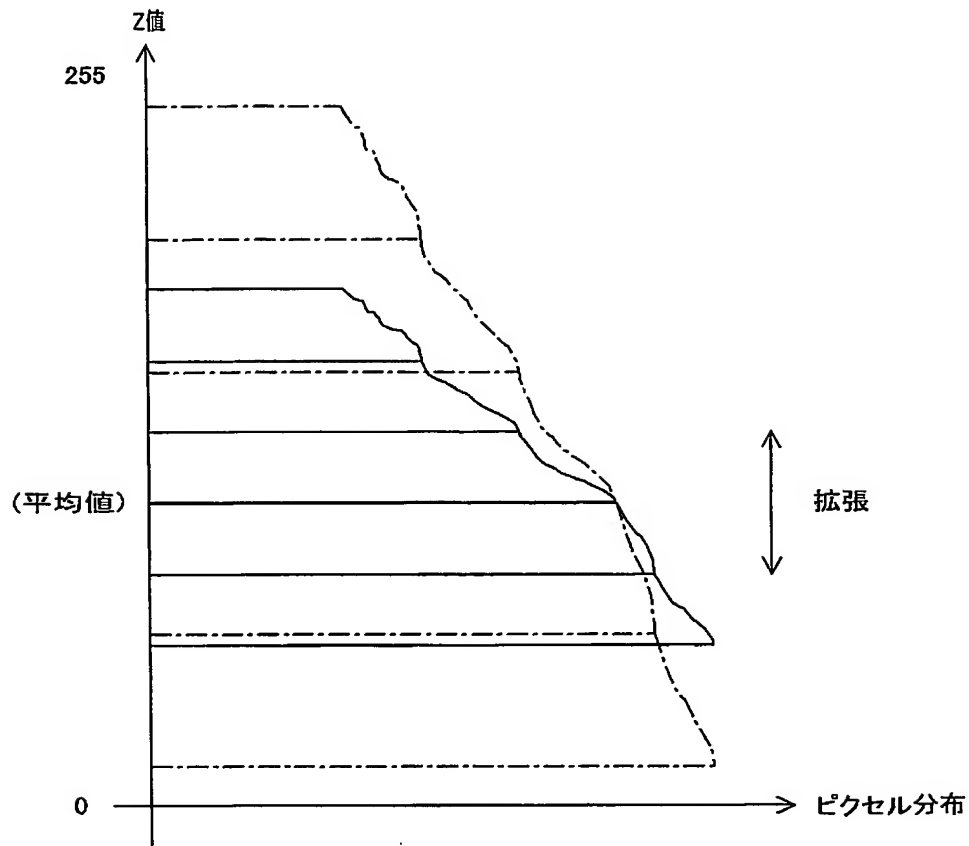
【図 14】



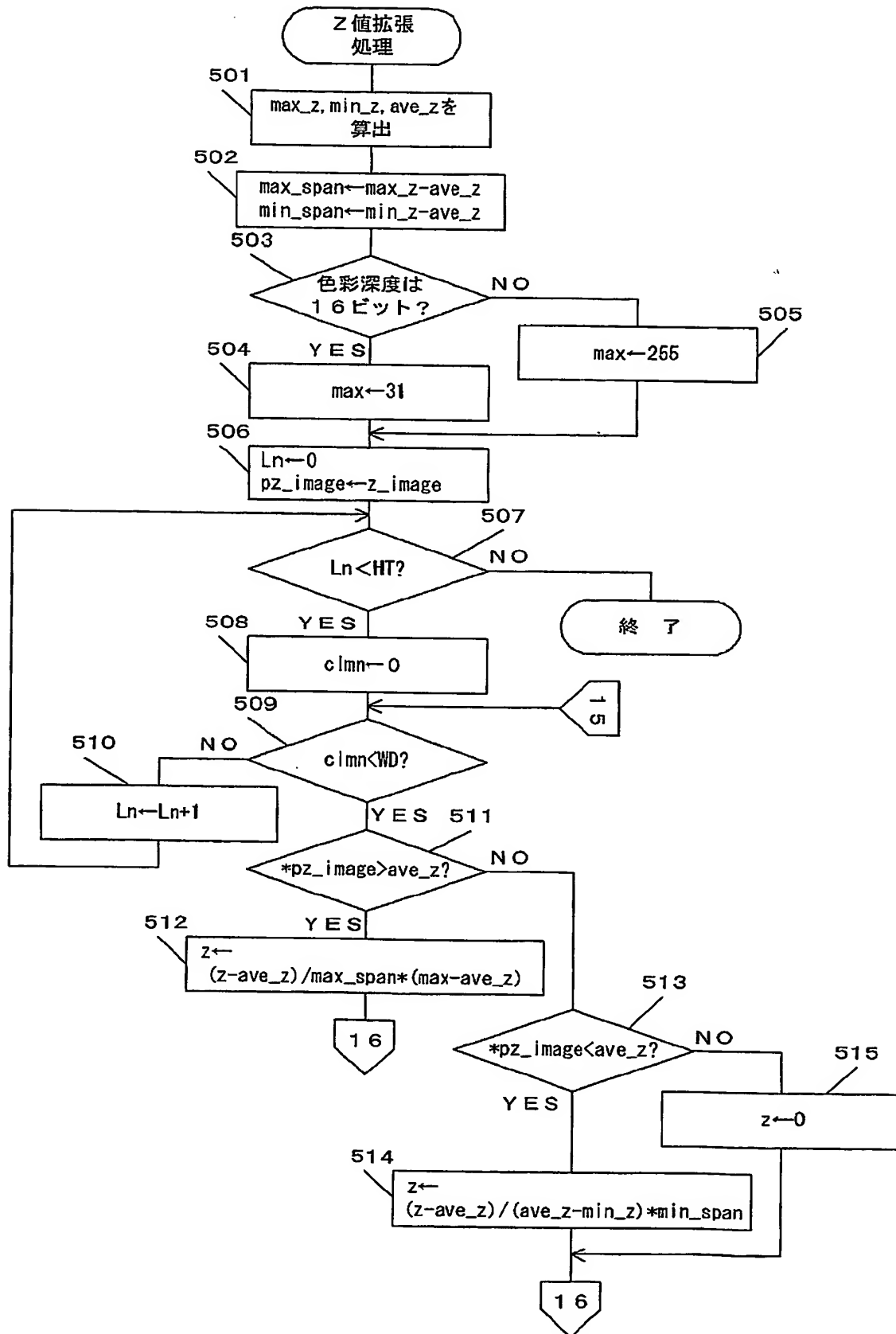
【図 15】



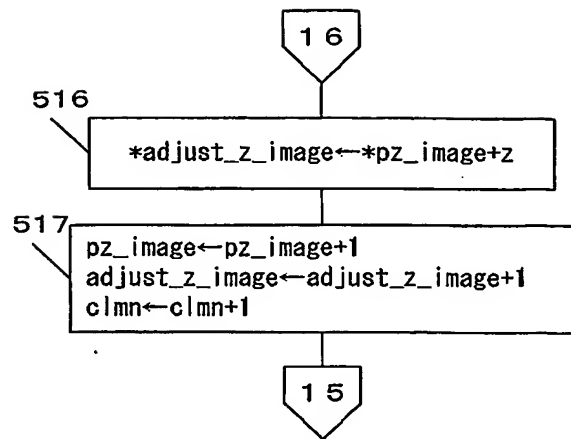
【図 16】



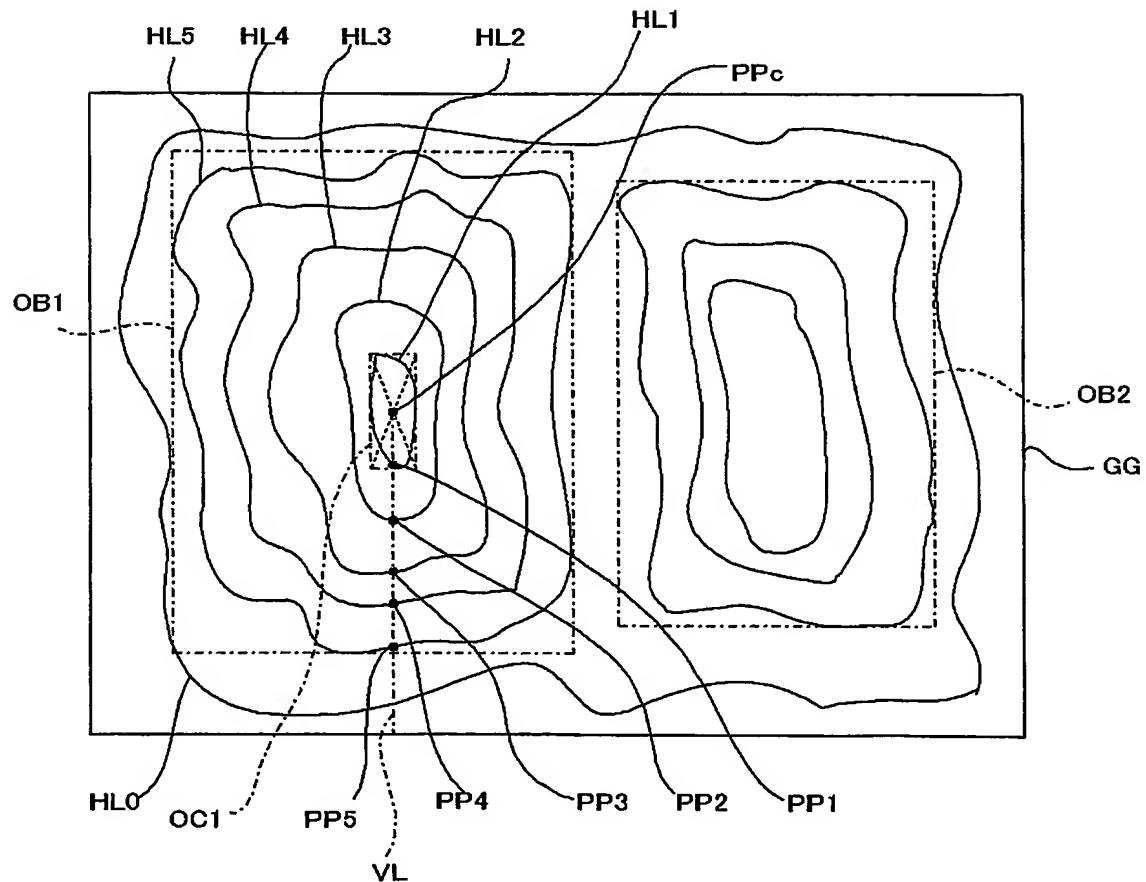
【図 17】



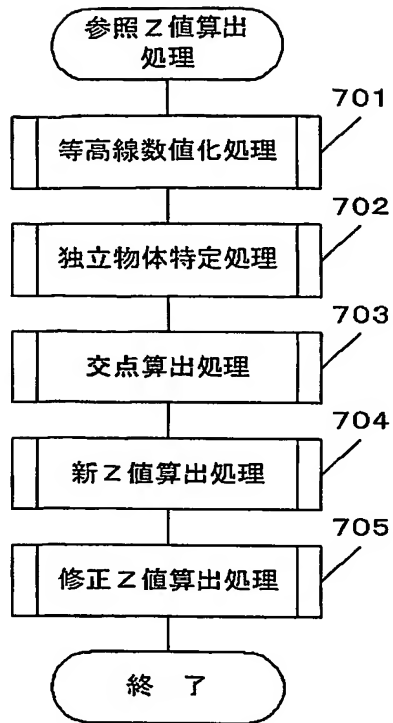
【図 18】



【図 19】



【図 20】

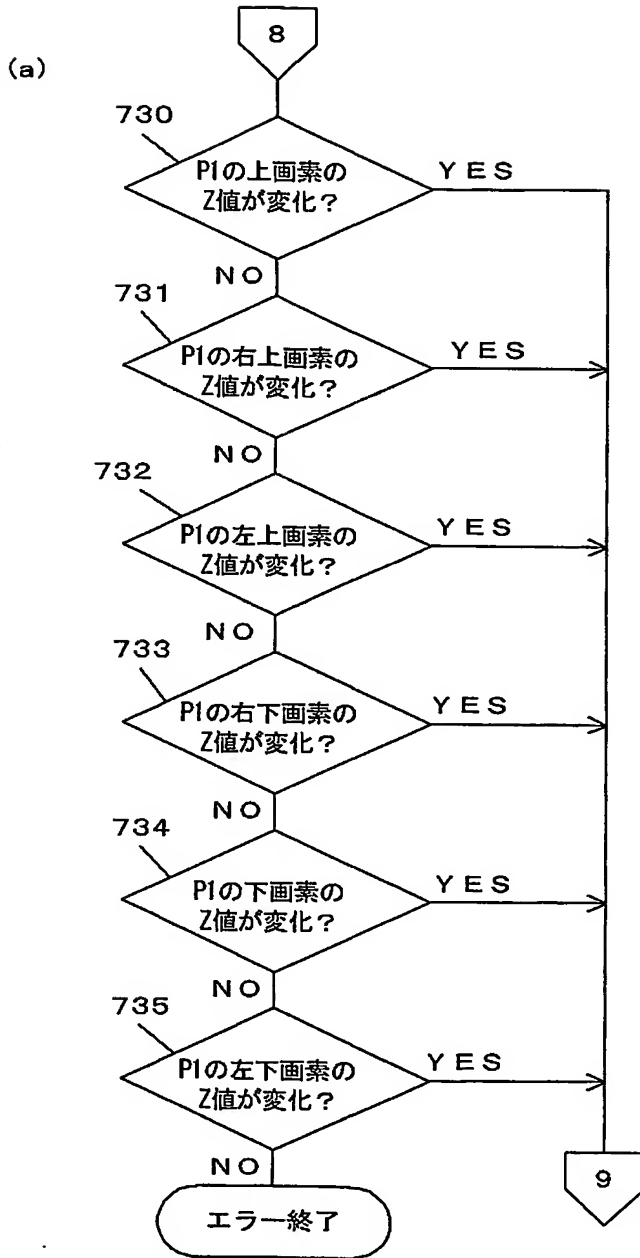


```

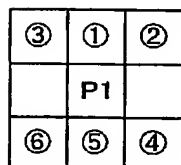
graph TD
    Start([等高線数値化  
処理]) --> 710
    710["max ← 255 / (丸め係数)  
Target_z ← 0"] --> 711
    711{"Target_z < max?"} -- NO --> End([終了])
    711 -- YES --> 712
    712["Ln ← 0"] --> 713
    713{"Ln < HT?"} -- NO --> 714
    714["Target_z ← Target_z + 1"] --> 711
    713 -- YES --> 715
    715["CLM ← 0  
P1 ← -1  
P2 ← -1  
fnd ← 0"] --> 716
    716{"CLM < WD?"} -- NO --> 723
    716 -- YES --> 717
    717{"*outline == Target_z?"} -- NO --> 721
    717 -- YES --> 718
    718{"P1 ≥ 0?"} -- YES --> 719
    718 -- NO --> 721
    719["P1 ← CLM"] --> 720
    720["CLM ← CLM + 1"] --> 716
    721{"P1 ≥ 0?"} -- YES --> 722
    721 -- NO --> 724
    722["P2 ← CLM  
fnd ← fnd + 1"] --> 720
    723{"fnd > 0?"} -- YES --> 721
    723 -- NO --> 724
    724["Ln ← Ln + 1"] --> 713

```


【図 22】

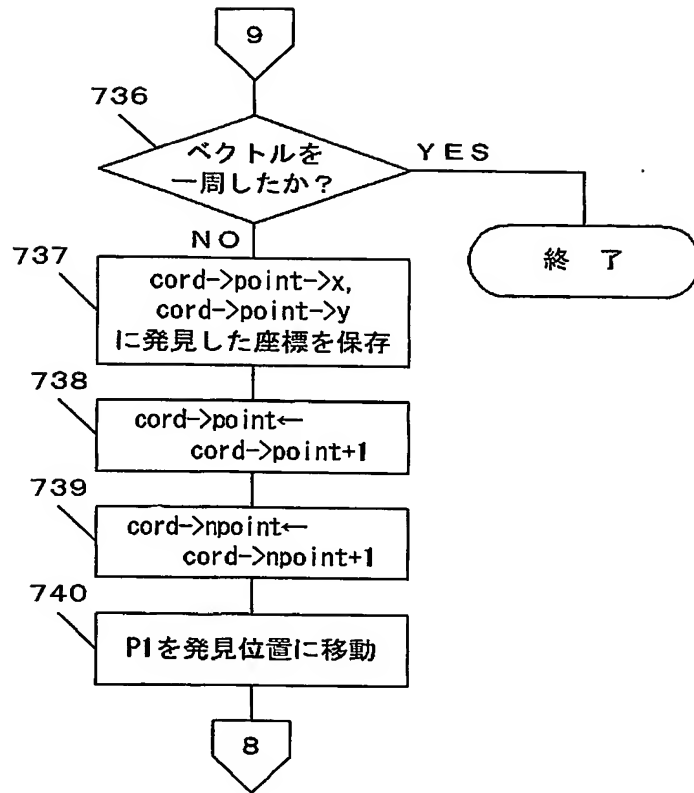


(b)

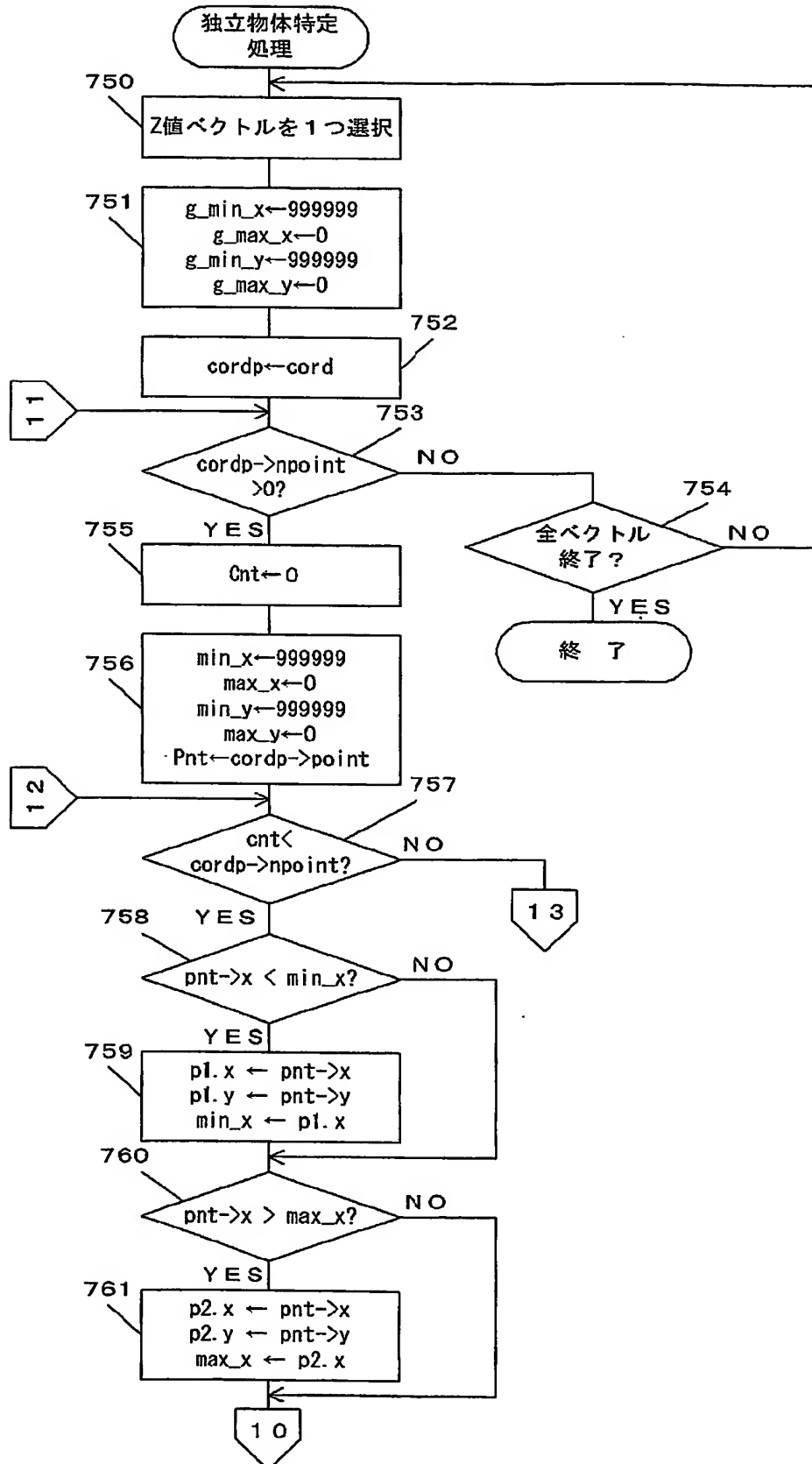


(Z値変化画素探索順序)

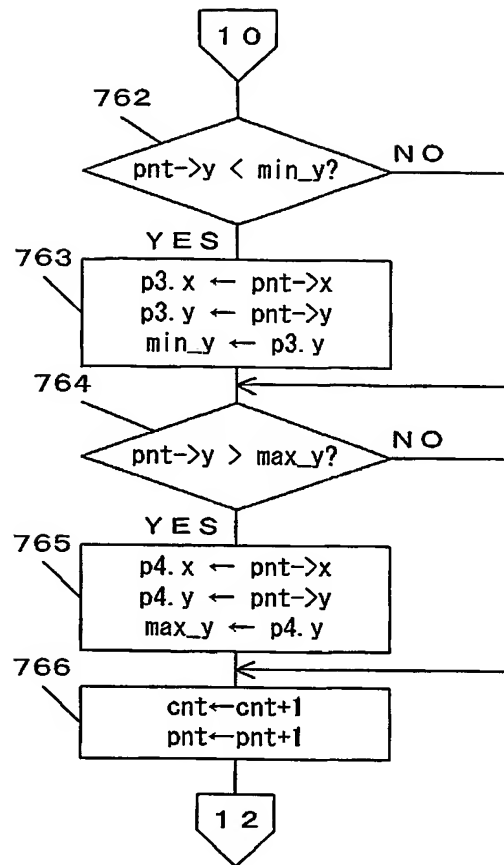
【図 23】



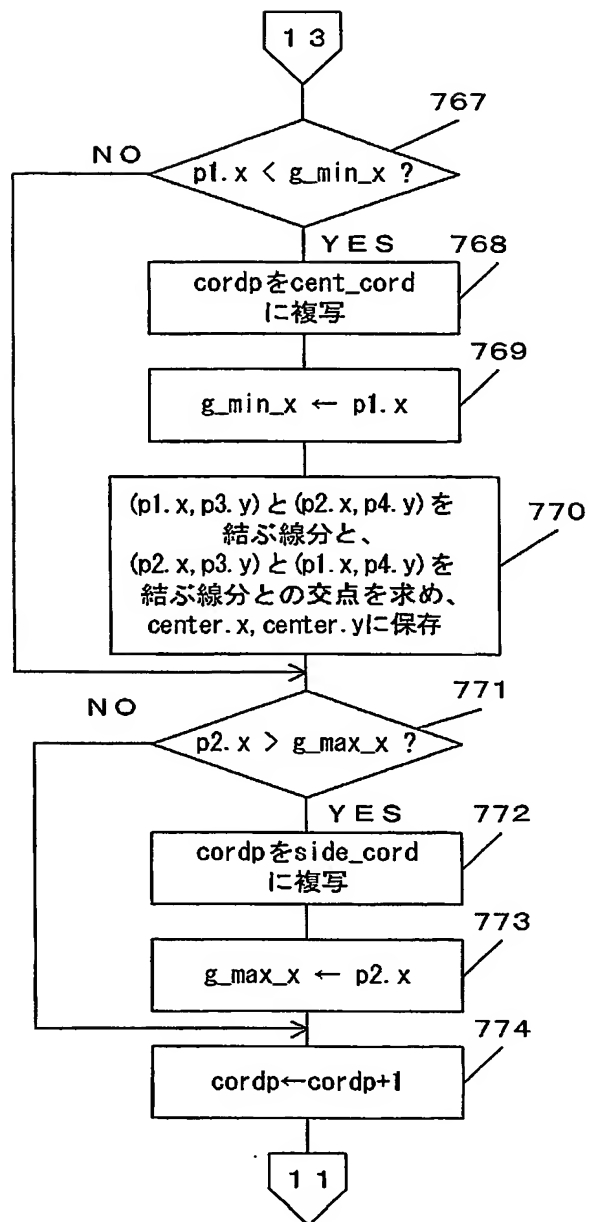
【図 24】



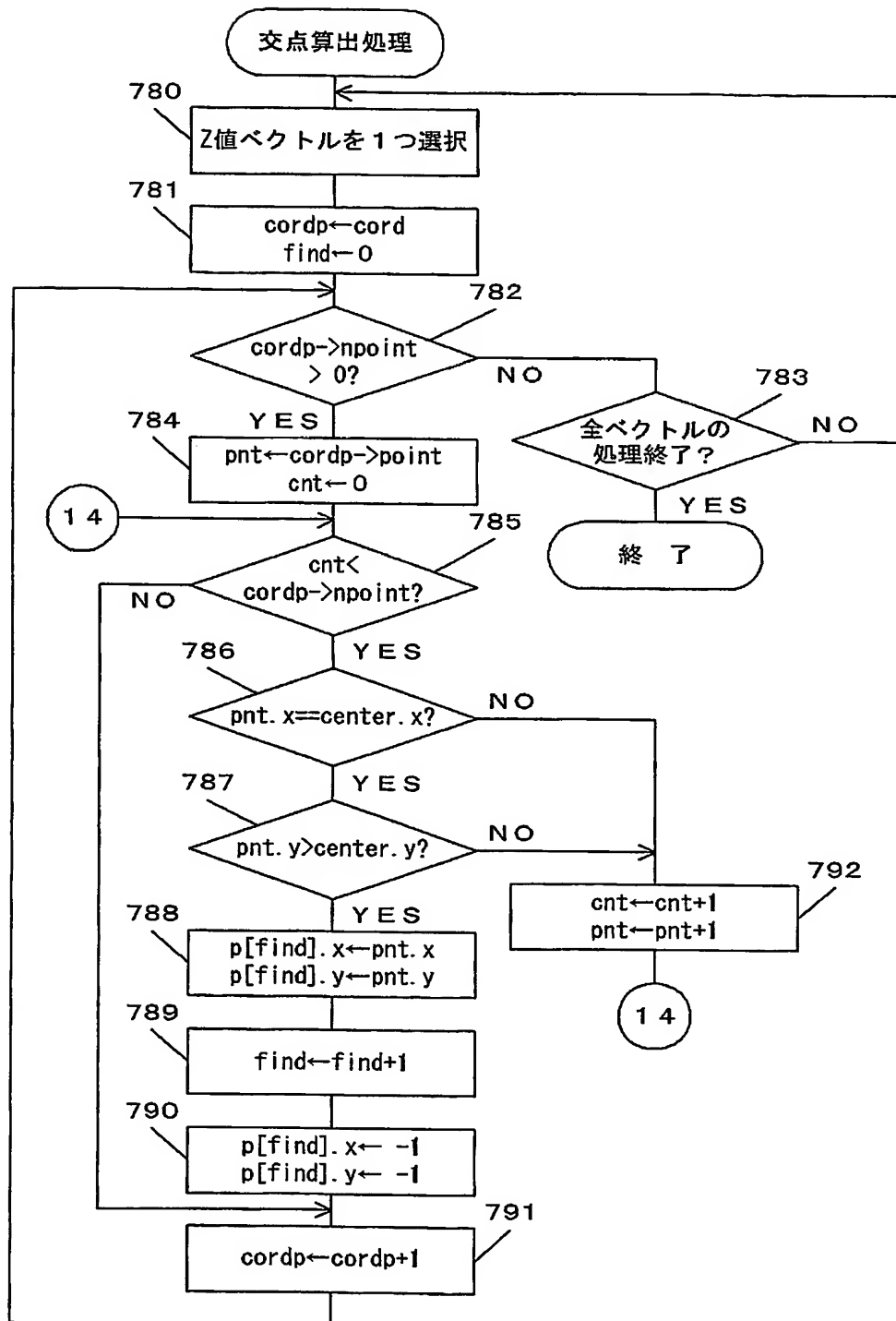
【図 25】



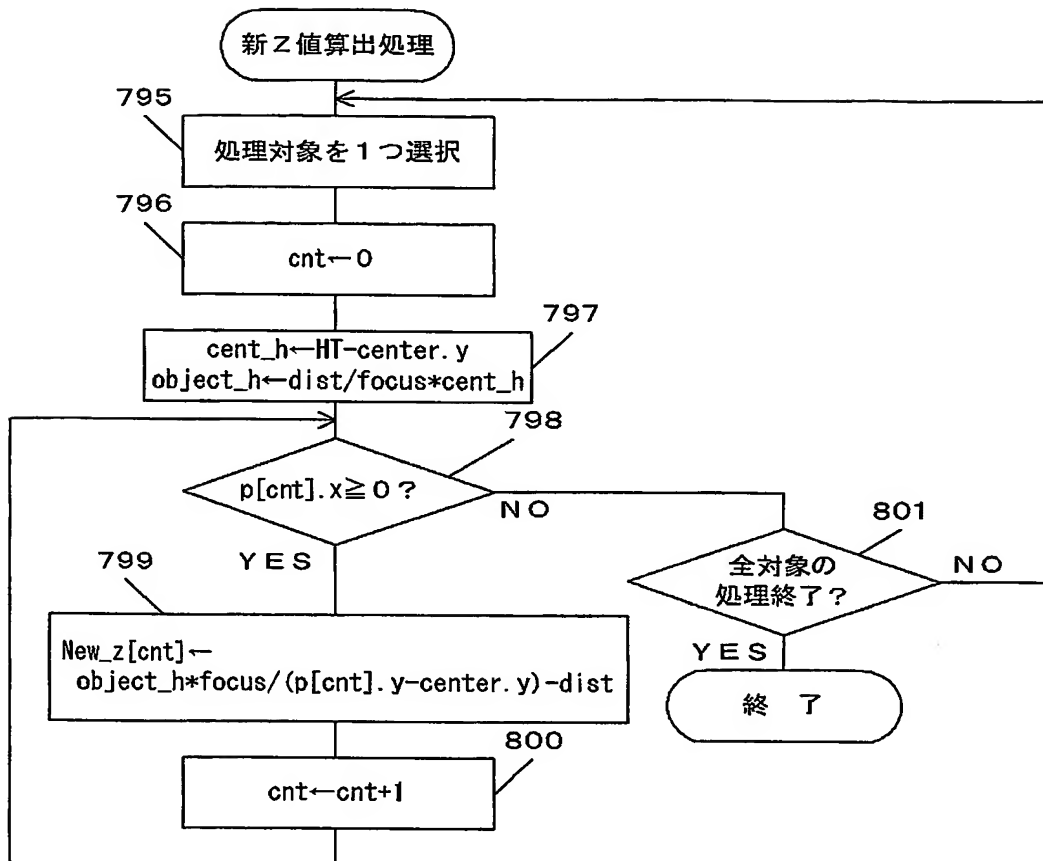
【図 26】



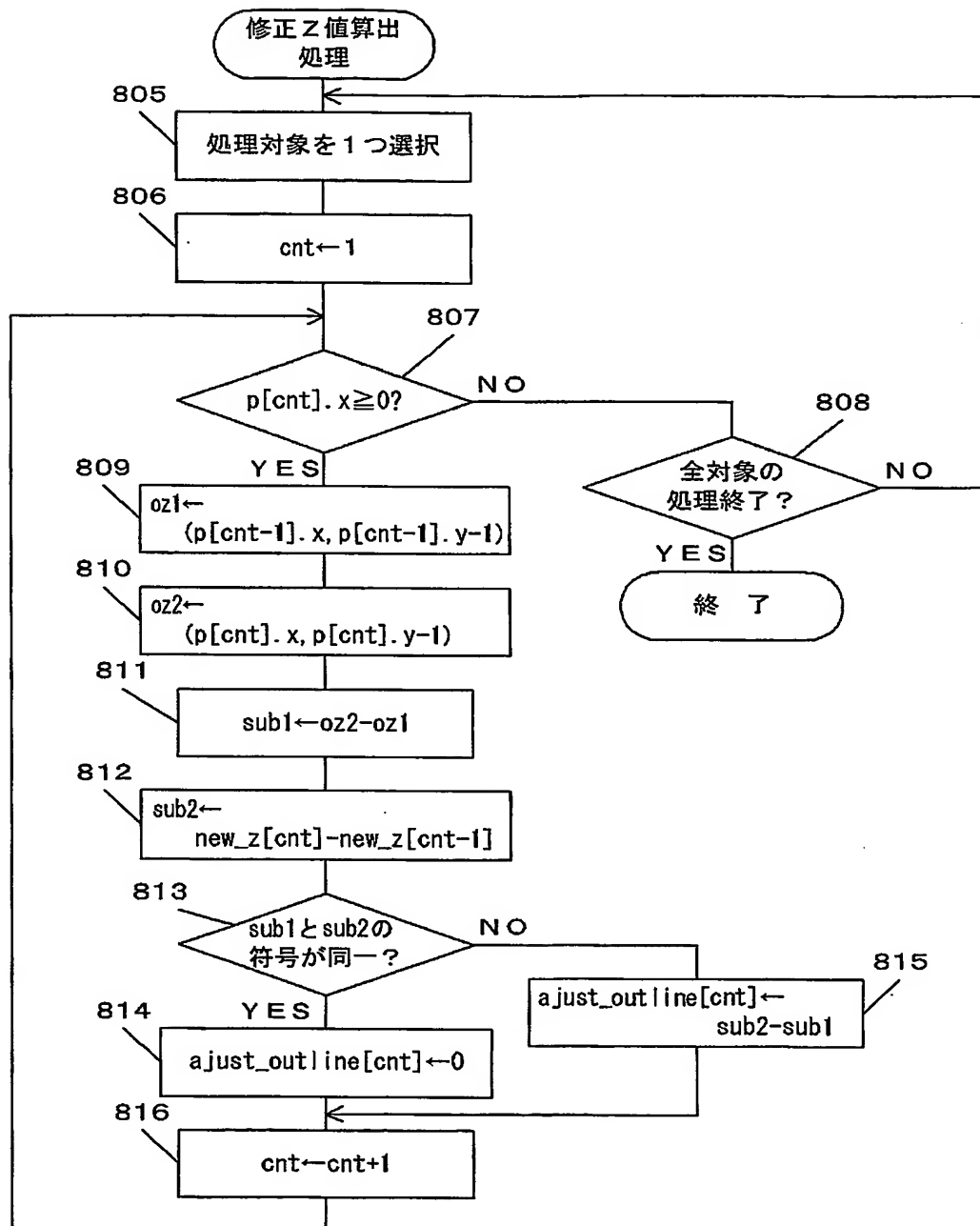
【図 27】



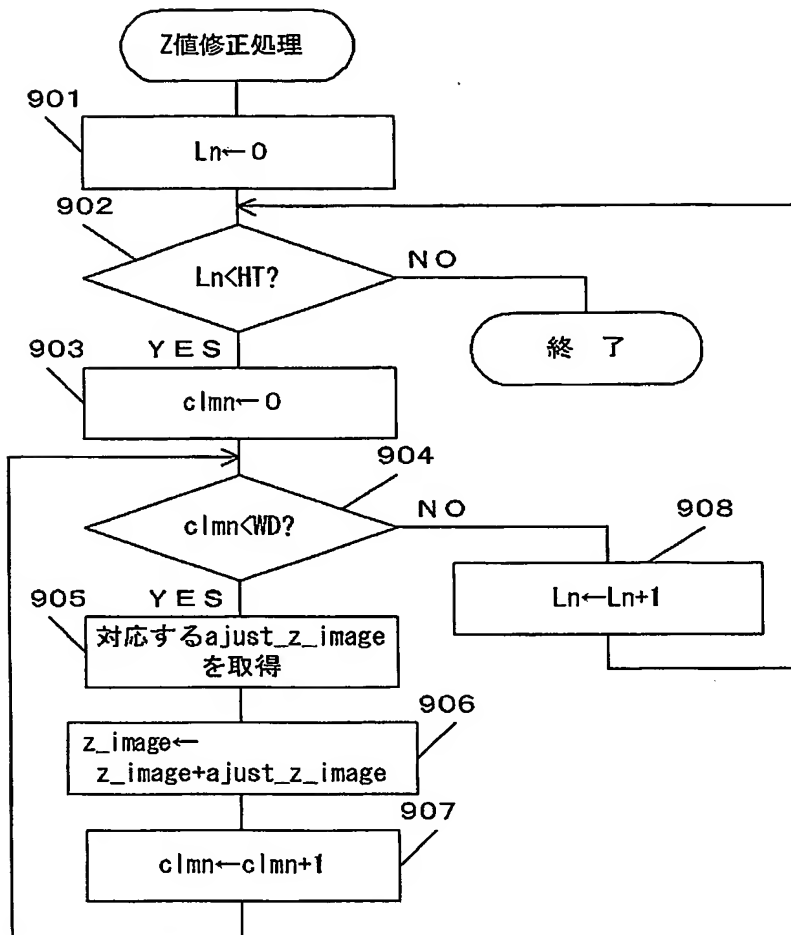
【図 28】



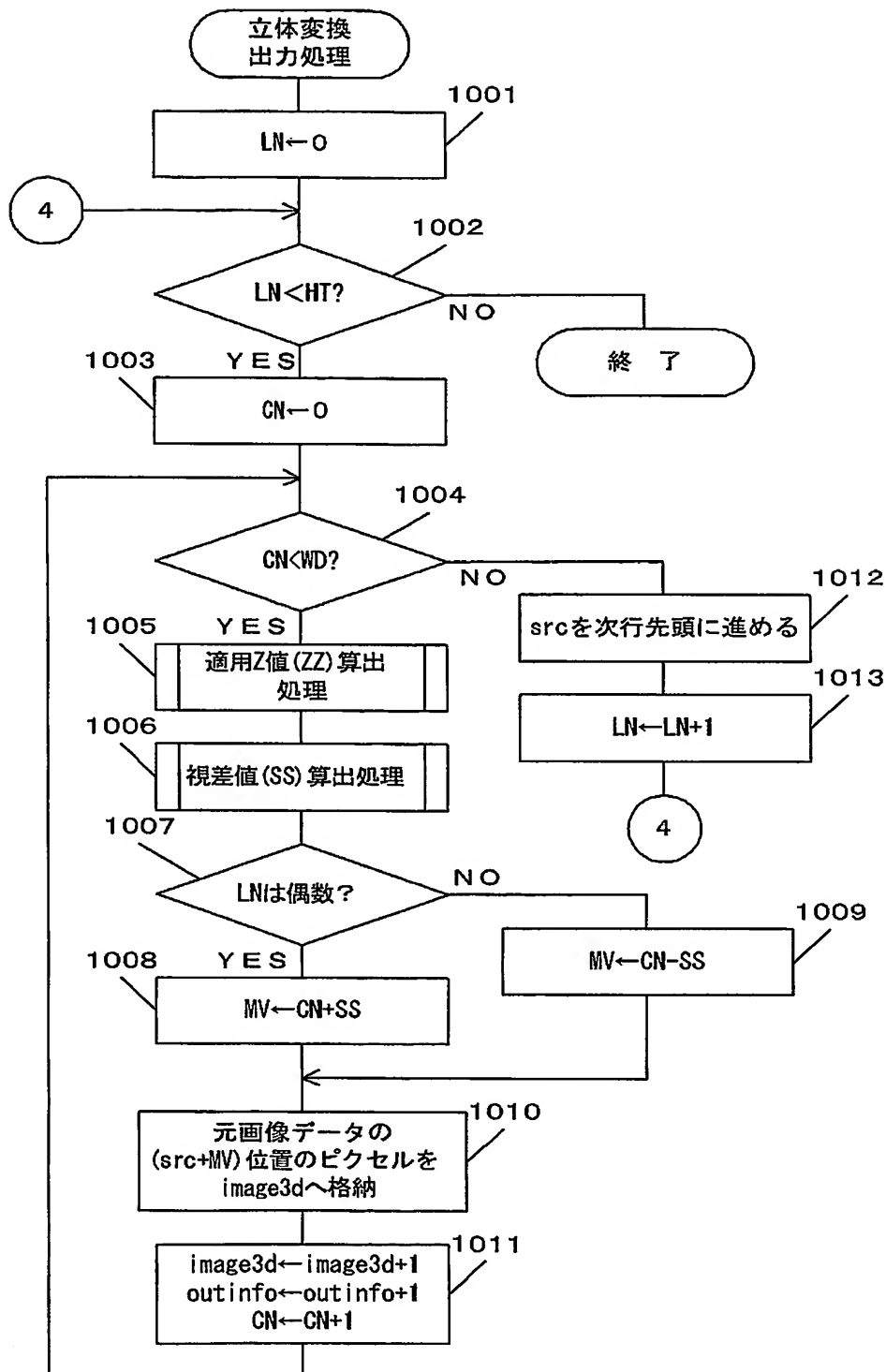
【図 29】



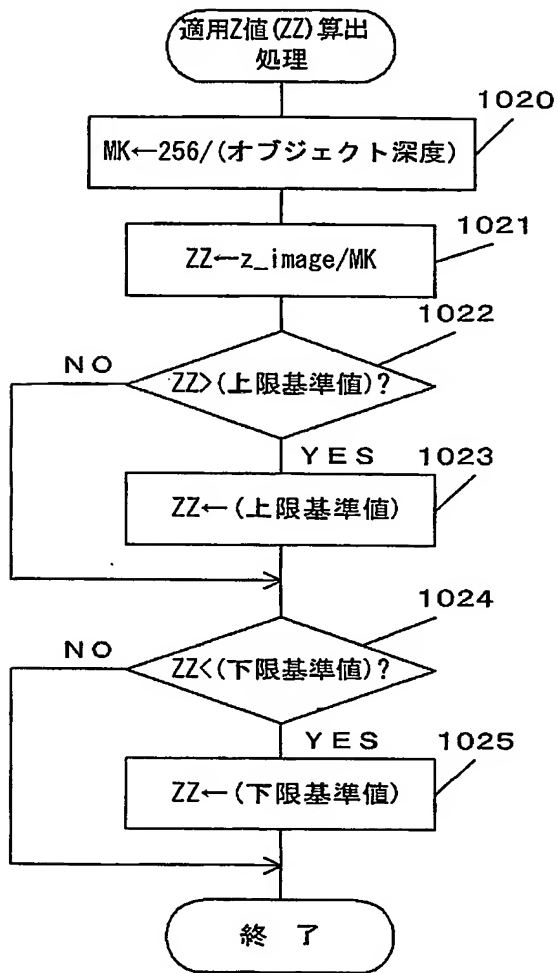
【図 30】



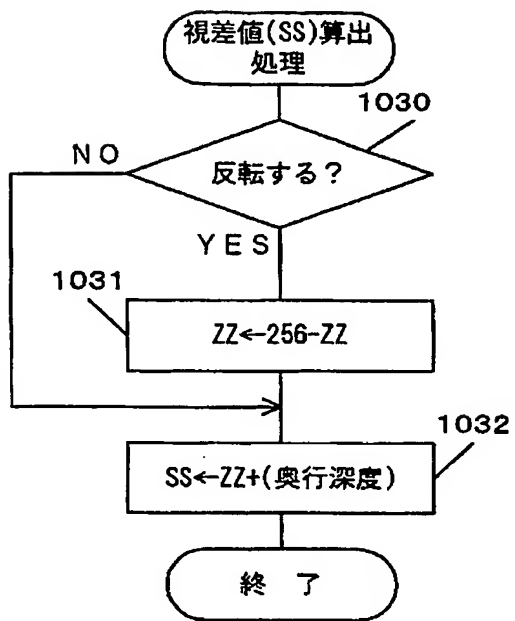
【図 31】



【図 3 2】

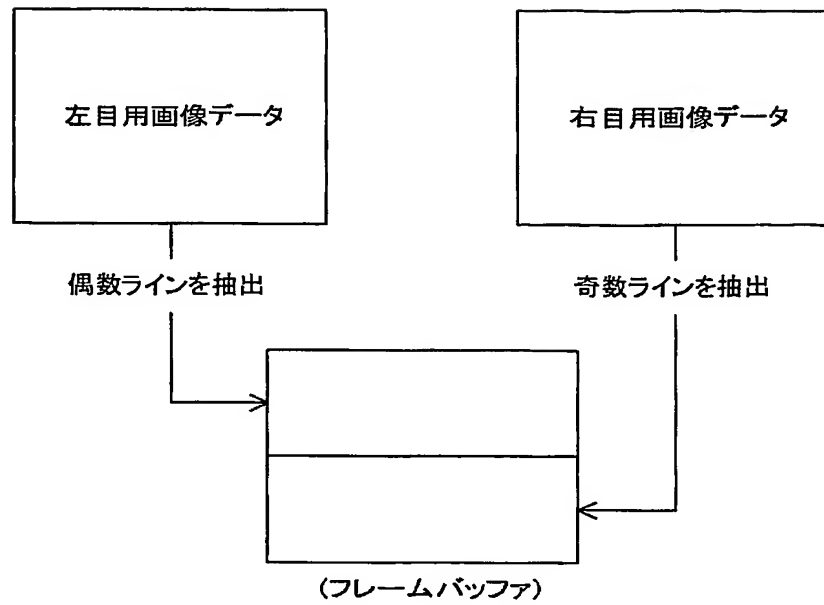


【図 33】

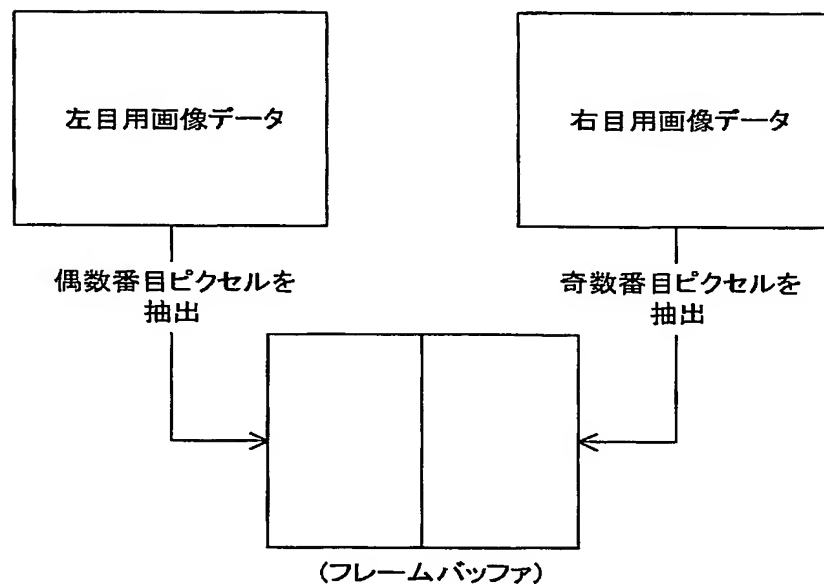


【図 3 4】

(a) 上下方式

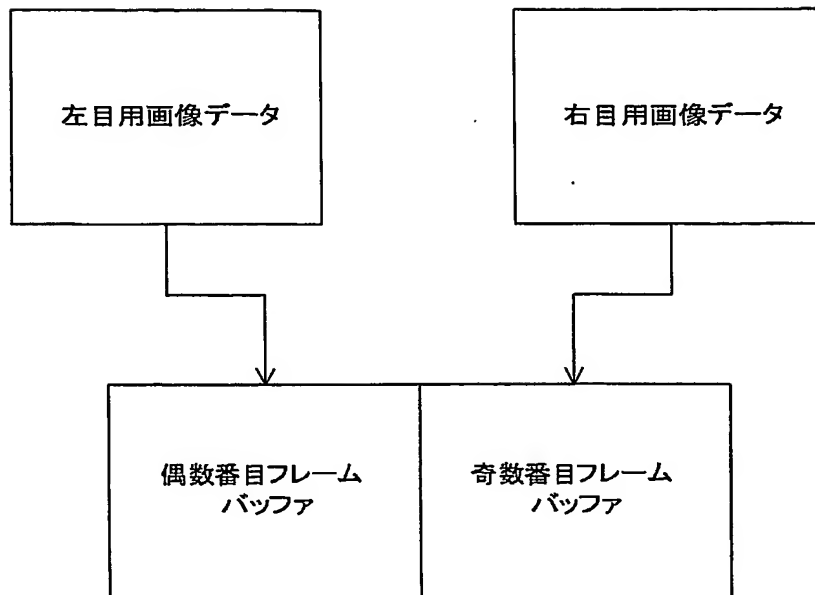


(b) 左右方式

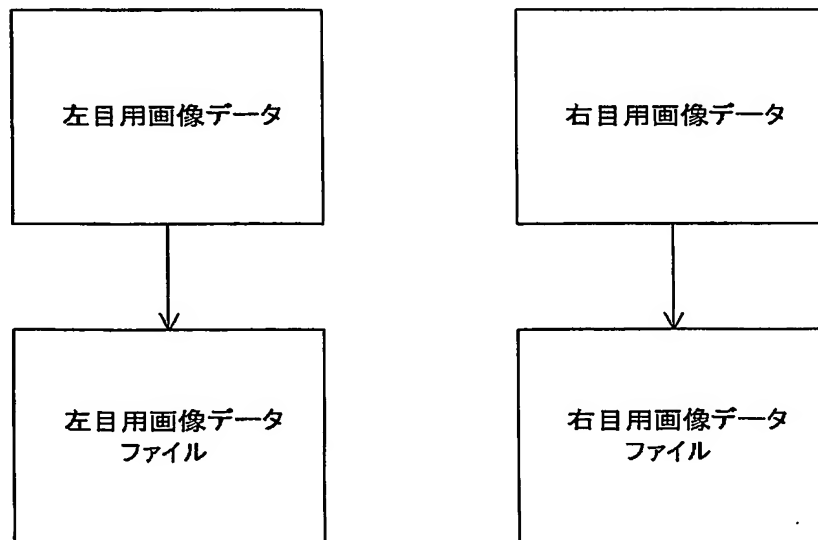


【図 3 5】

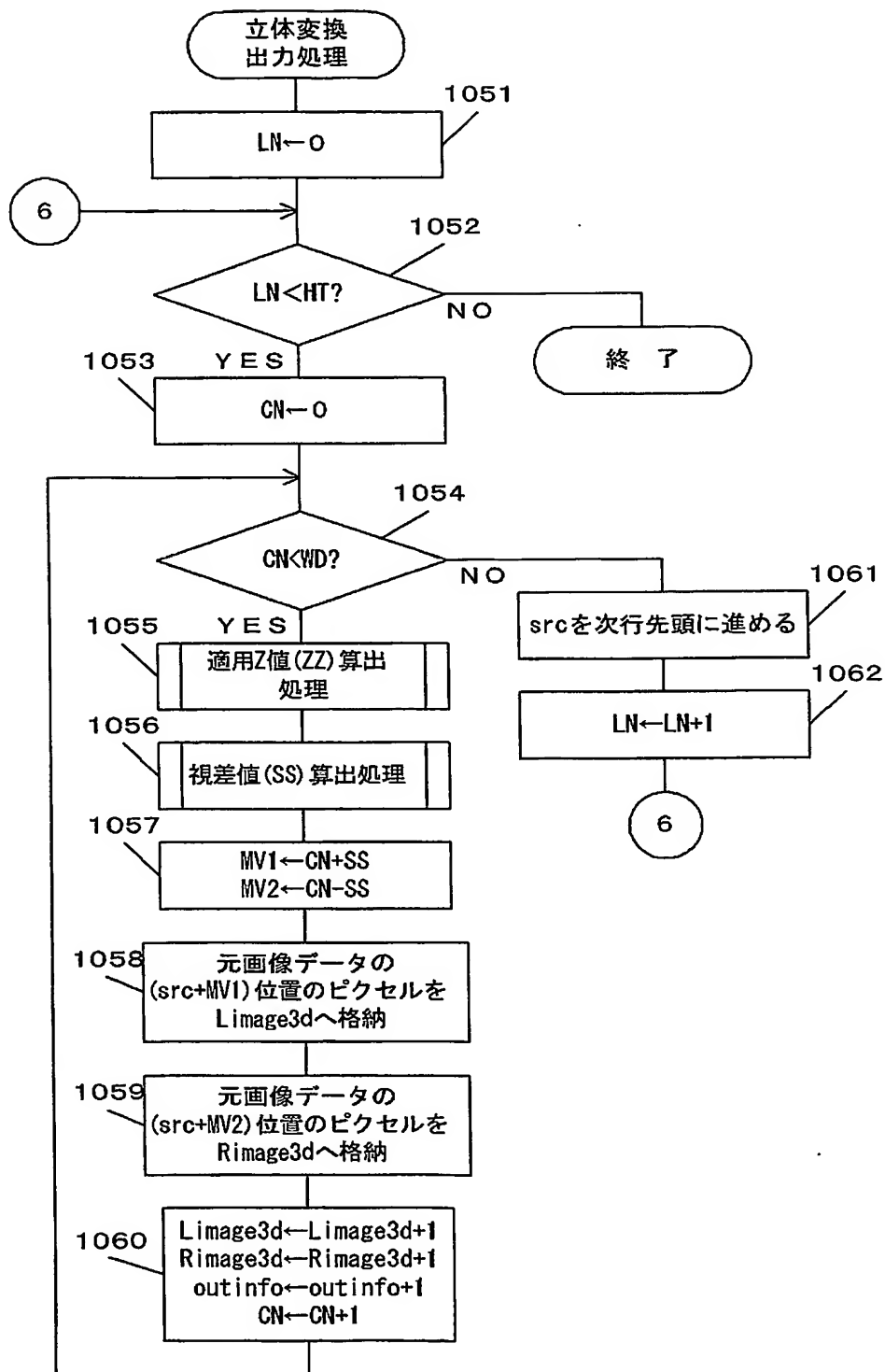
(a) 2フレーム方式



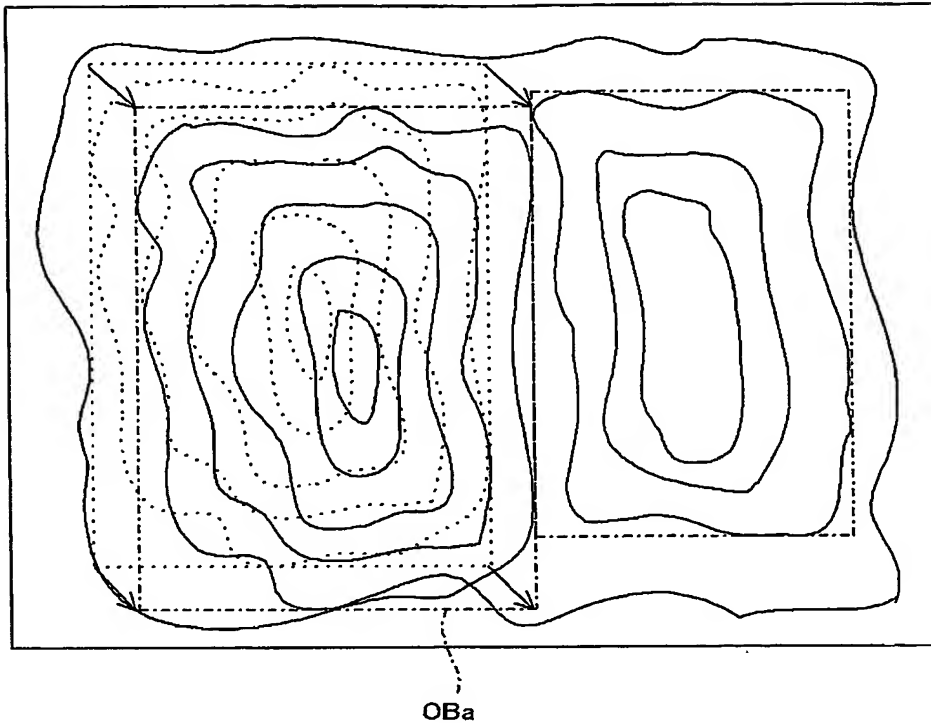
(b) Odd-Evenファイル分離方式



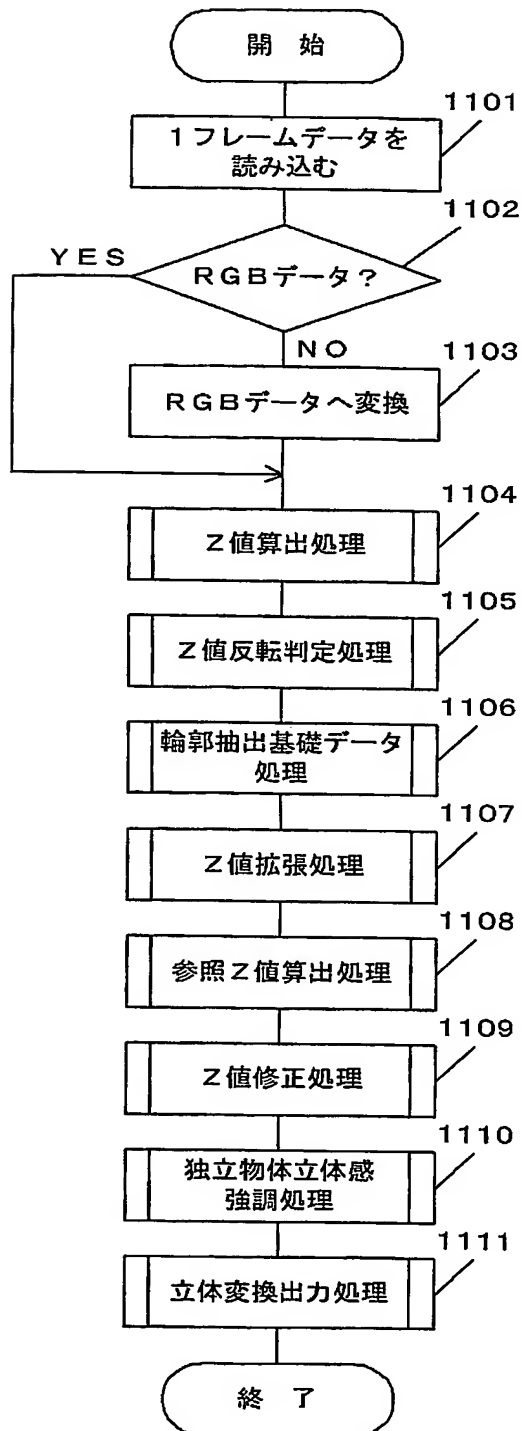
【図 36】



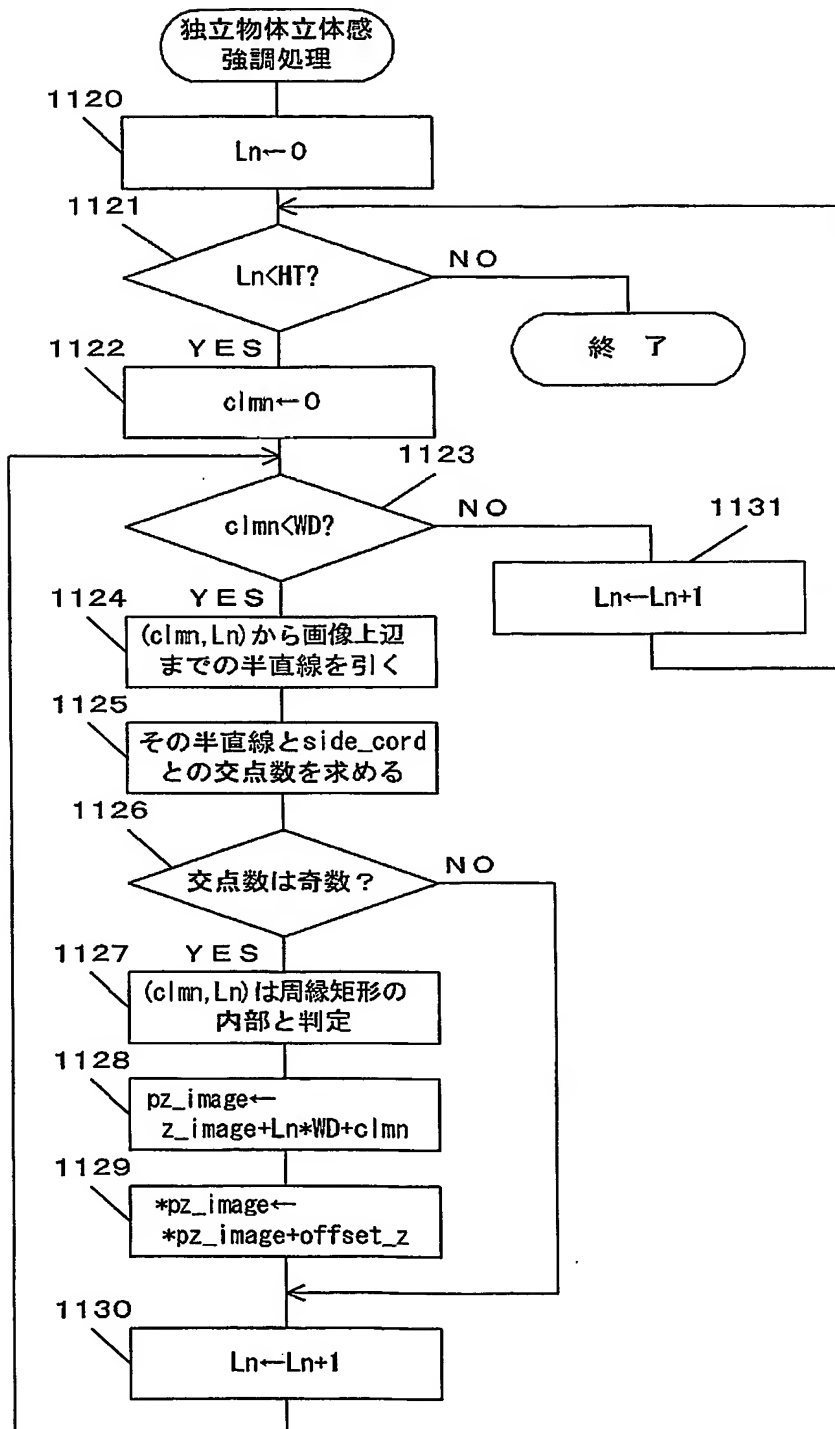
【図 37】



【図 38】



【図 39】



【書類名】 要約書

【要約】

【課題】 二次元カラー画像データに基づいて、自然な立体視を得ることができるようにした立体視画像データ作成方法および立体視画像データ作成装置を提供することを目的としている。

【解決手段】 画像データのグレースケール値を用いて奥行き値（Z値）を算出し、輪郭抽出、参照Z値算出等の複数の修正処理を行うことで、立体視画像データに与える視差値を適切な値にしており、結果的に、自然な立体感を得ることができるという効果を得る。

【選択図】 図6

特願 2 0 0 2 - 2 3 8 7 0 9

出 願 人 履 歷 情 報

識別番号

[5 9 6 0 3 0 7 3 7]

1. 変更年月日

1 9 9 6 年 1 月 3 1 日

[変更理由]

新規登録

住 所

千葉県柏市松葉町 4 - 1 - 1 1 - 2 0 1

氏 名

江良 一成

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.